SEST-6577
Geographic Information Systems for Security Studies
Lab 08 (+ Problem Set 8)

Yuri M. Zhukov
Associate Professor
Georgetown University

October 24, 2024

Goal: analyze flood risk models in New Orleans

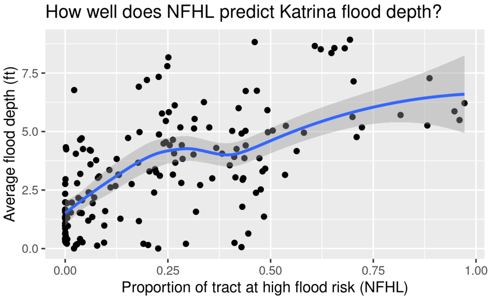


Figure 1: Predicting catastrophic events

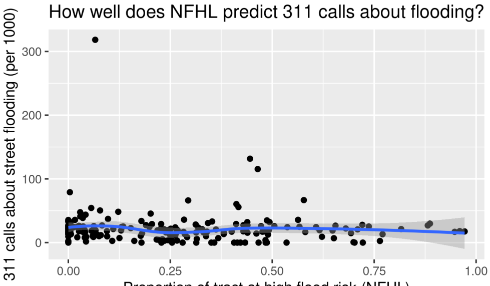


Figure 2: Predicting everyday events

Case study: Hurricane Katrina was a Category 5 storm that hit New Orleans in 2005

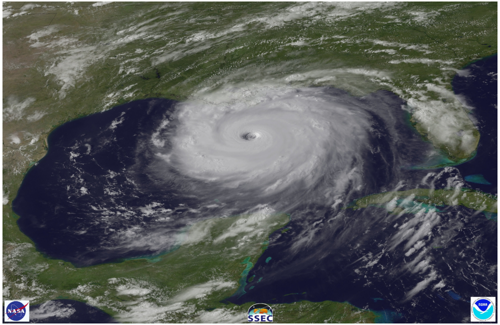


Figure 3: Direct hit

Figure 4: Levees break

It claimed over 1800 lives and over $100 billion in property damage

Figure 5: French Quarter

Figure 6: Lower 9th Ward

We will examine the geographic distribution of post-Katrina flooding in NOLA

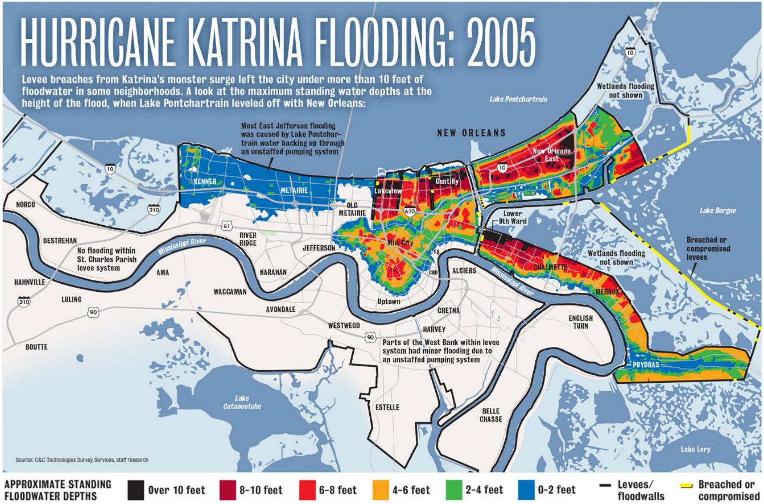


Figure 7: We will use these data, among others

We will look at whether communities of color were more vulnerable to flooding

Figure 8: Legacy of housing discrimination

We'll also evaluate the accuracy of the Federal Emergency Management Agency's (FEMA) National Flood Hazard Layer (NFHL) model in predicting (a) flood depth and (b) non-emergency municipal service calls (3-1-1 calls) about flooding

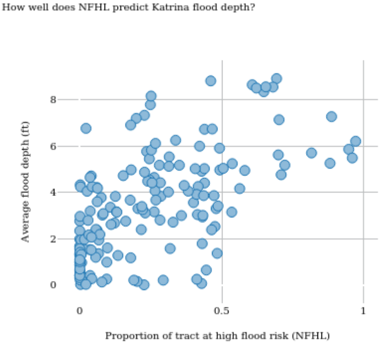


Figure 9: Predicting the past

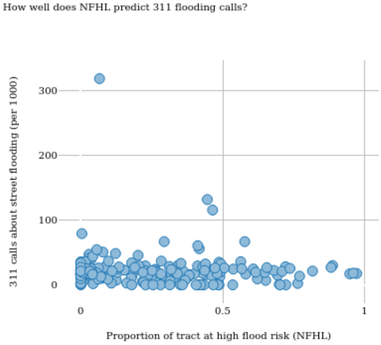


Figure 10: Predicting the everyday

Overview of lab exercise and problem set

1. Lab exercise
   a) Hands-on experience analyzing and editing raster data
   b) Calculate zonal statistics of flood depth for Census tracts
   c) Re-classify and subset flood risk raster data
   d) Integrate flood data with data on housing discrimination, race, 311 calls
2. Problem set
   a) Create statistical graphics (scatterplots) evaluating performance of flood hazard model in predicting:
      - Katrina flood levels
      - per capita 311 calls about street flooding

You can make these plots in QGIS or in R. Instructions for both are below

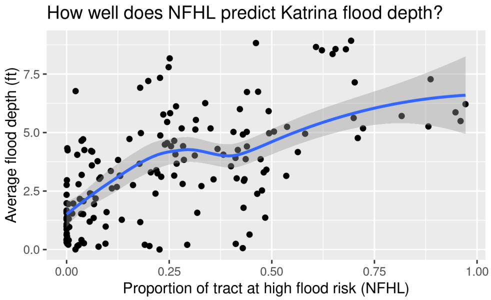


Figure 11: Scatterplot 1 in R

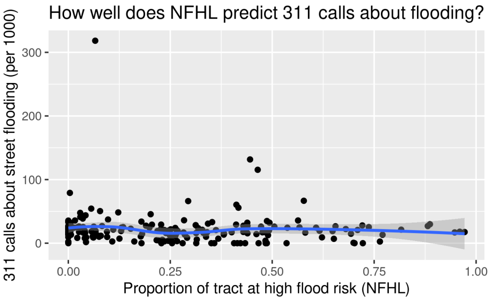


Figure 12: Scatterplot 2 in R

We have five sources of data:

Category	Type	Format	Data source
Hurricane Katrina flood depth	Raster	`.tif`	`NOAA`
2000 Census Tracts	Vector (polygon)	`.geojson`	`IPUMS NHGIS`
HOLC Redlining Maps	Vector (polygon)	`.geojson`	`Mapping Inequality`
National Flood Hazard Layer	Raster	`.tif`	`FEMA`
311 Calls	Table (geocoded)	`.csv`	`New Orleans Open Data`

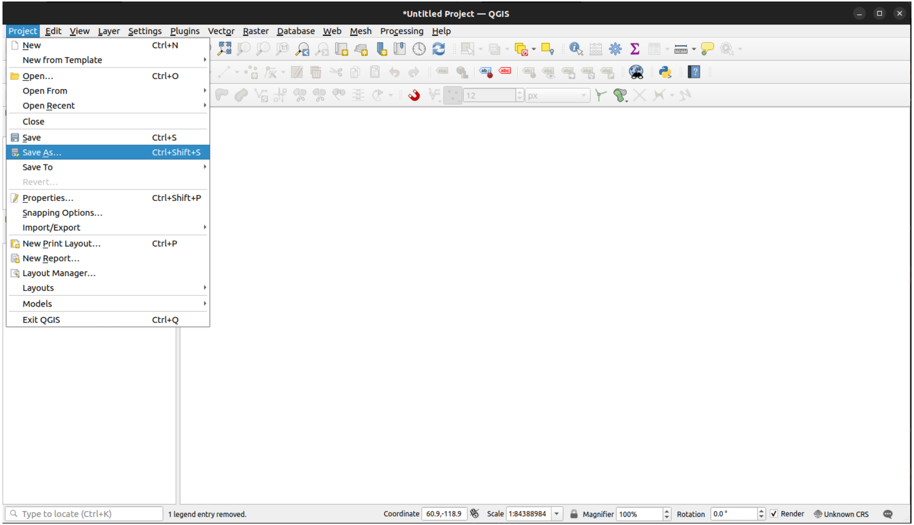
These are all in the `Lab08PS08.zip` file posted on Canvas.

Let's open QGIS. . .

QGIS

Always save your progress!

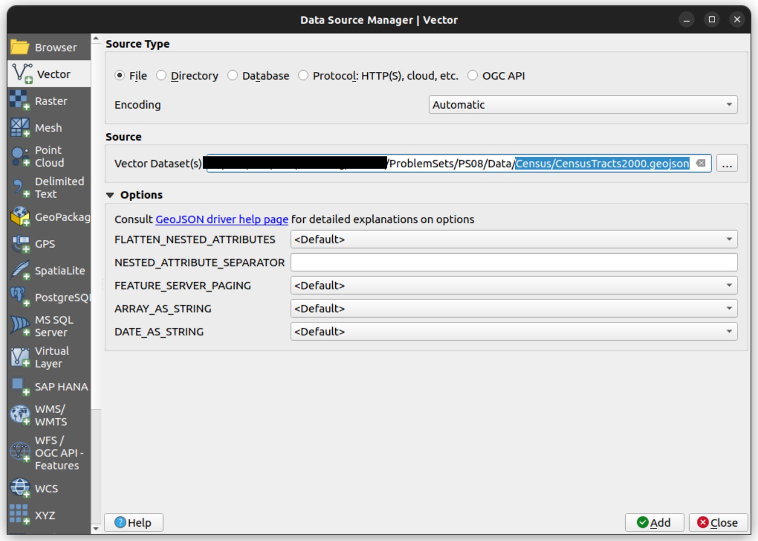
Go to Project → Save As...



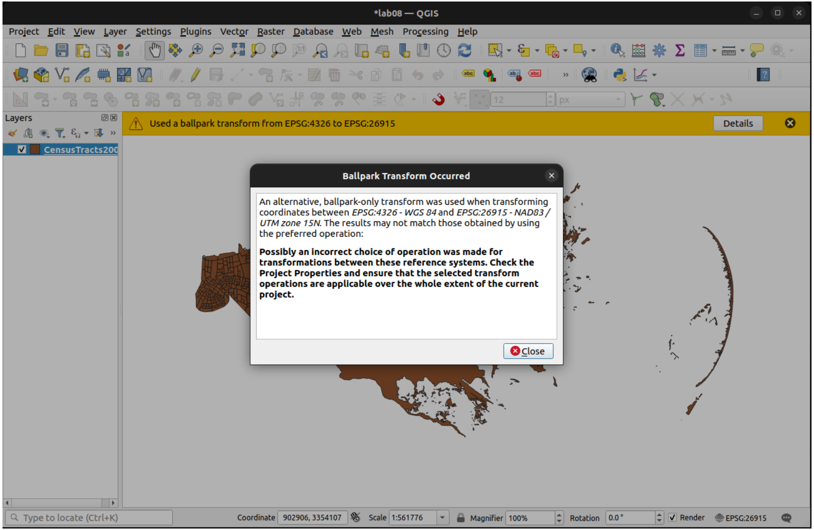
Zonal statistics

Let's begin by *calculating average flood depth in census tracts*.

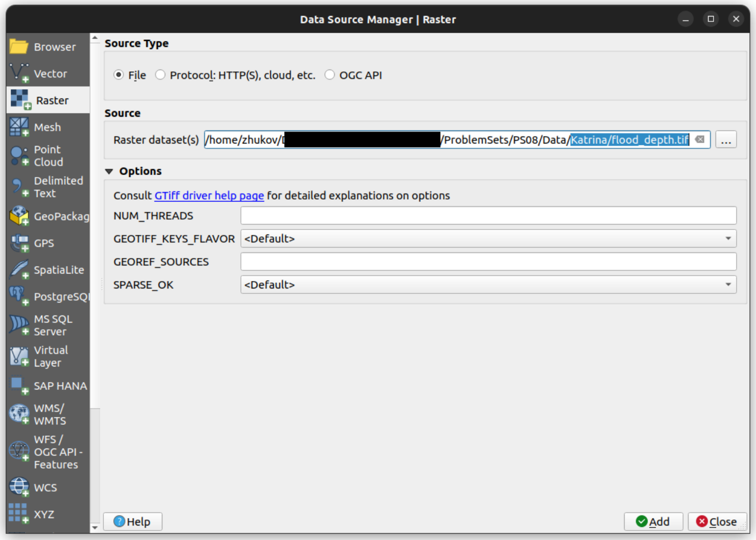
Load the `CensusTracts2000.geojson` (vector) from the `Data/Census` directory



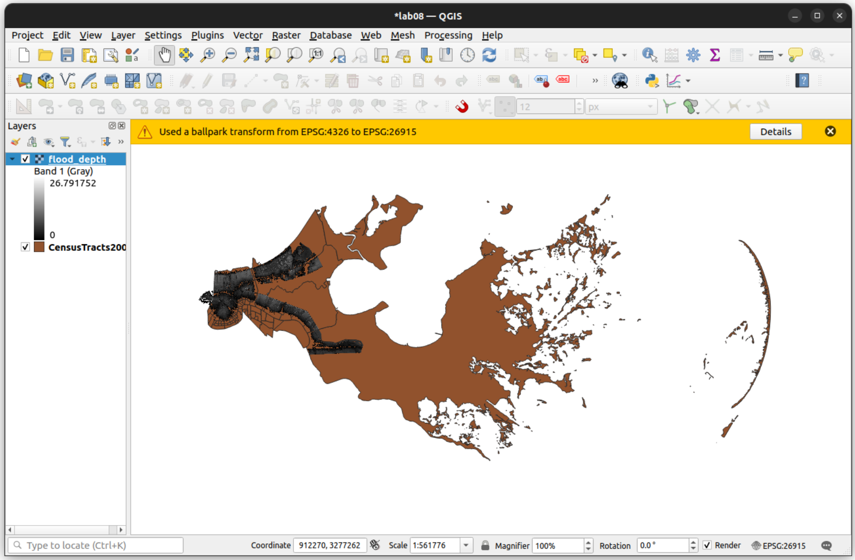
You may receive a “Ballpark Transform Occurred” warning. This sometimes happens when re-projecting from WGS84 (EPSG:4326, QGIS default) to UTM (EPSG:26915)



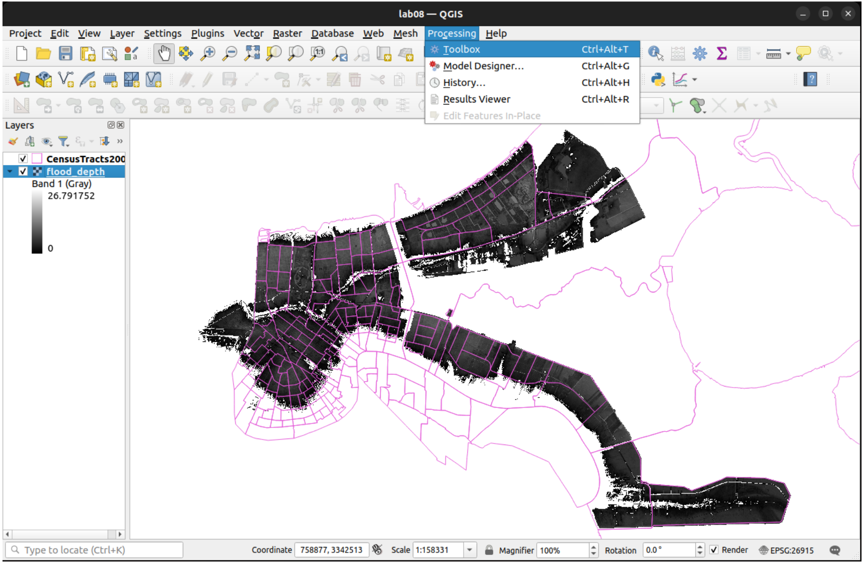
Load the raster `flood_depth.tif` from the `Data/Katrina` directory



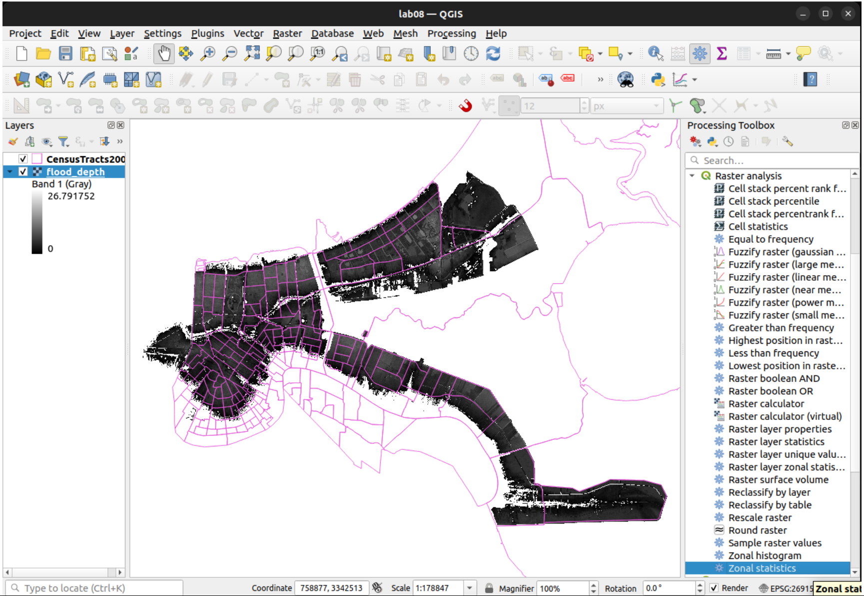
Note that these layers include both New Orleans and neighboring St. Bernard Parish



Let's calculate flood stats for each census tract. Open the `Processing Toolbox`



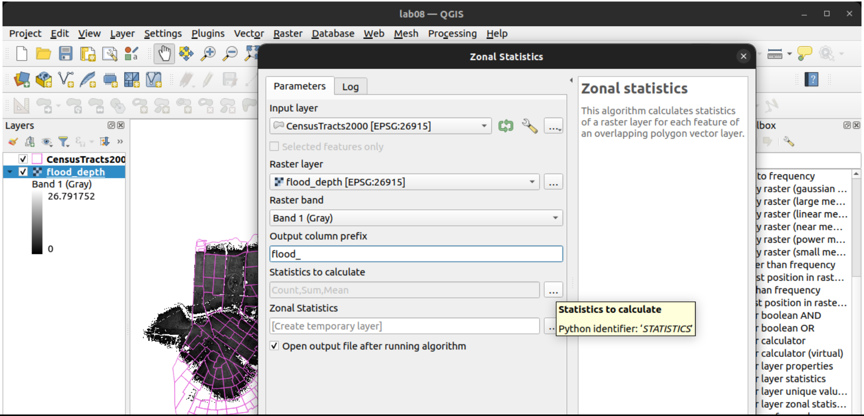
In the Toolbox, open `Raster analysis` submenu → `Zonal statistics` tool



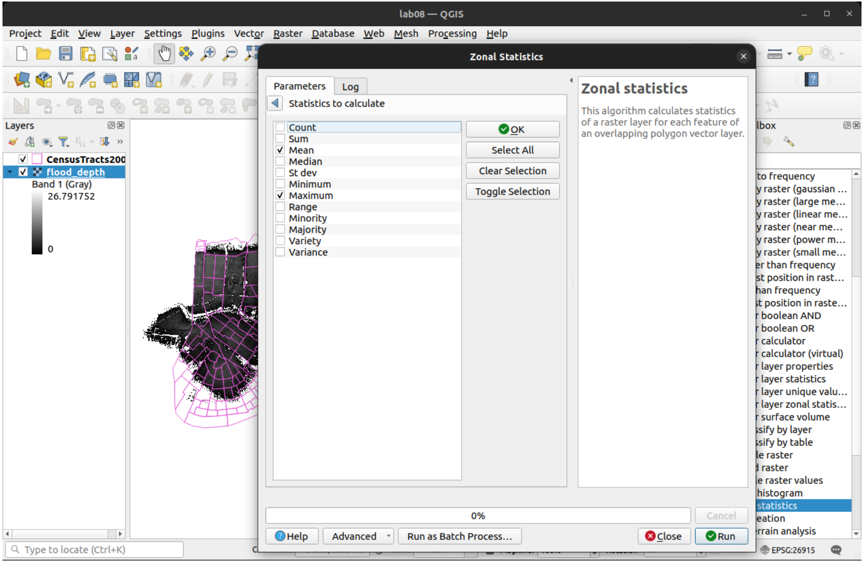
On the next screen, set

- Input layer = CensusTracts2000
- Raster layer = flood_depth
- Output column prefix = flood_

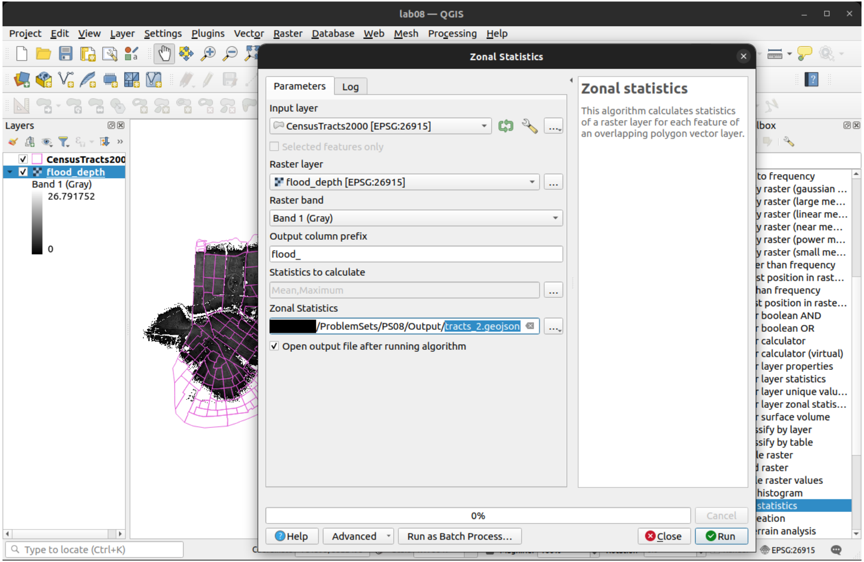
Click the [...] box next to Statistics to calculate



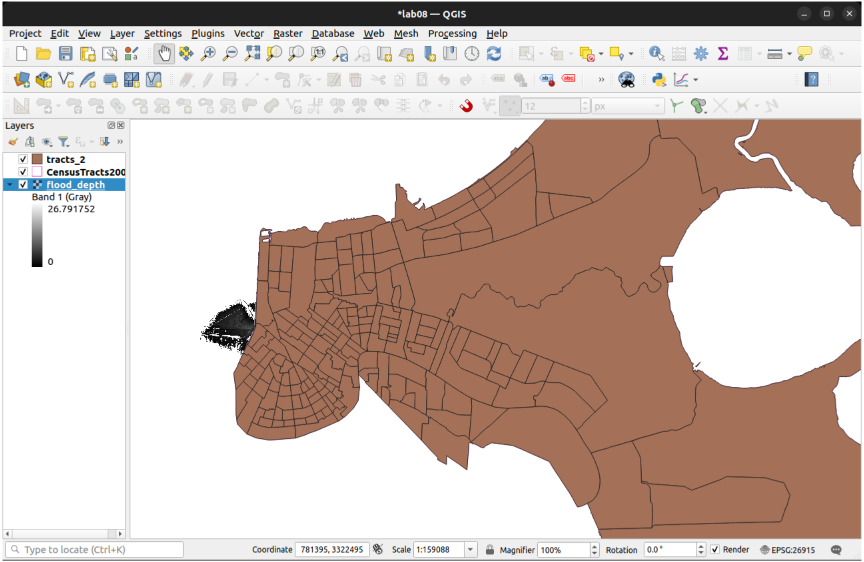
Select ✓ Mean, ✓ Maximum. Click OK



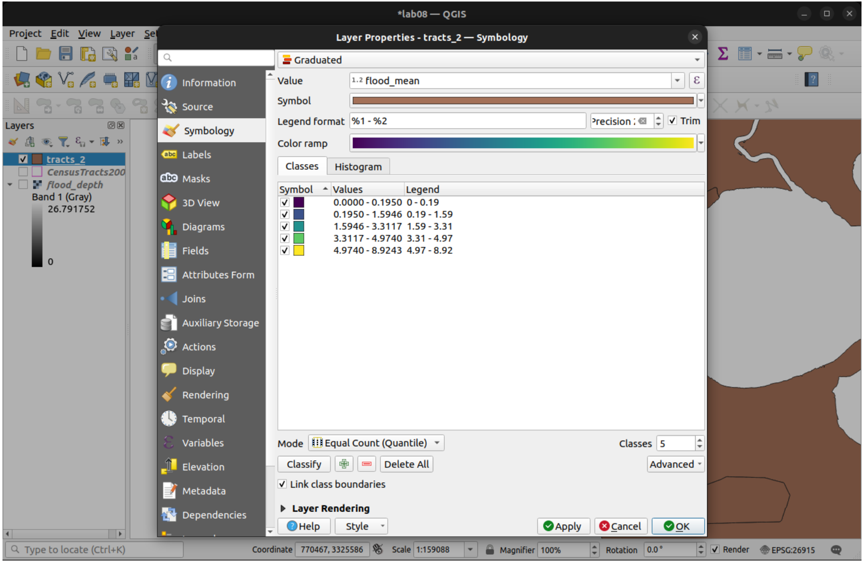
Save the output to file as `tracts_2.geojson`. Click `Run`



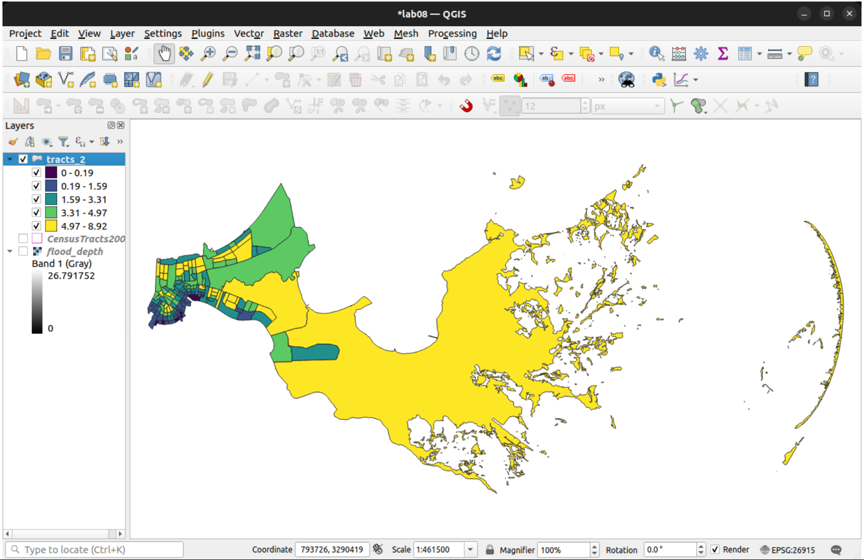
This will add a new layer, `tracts_2`, to your project window



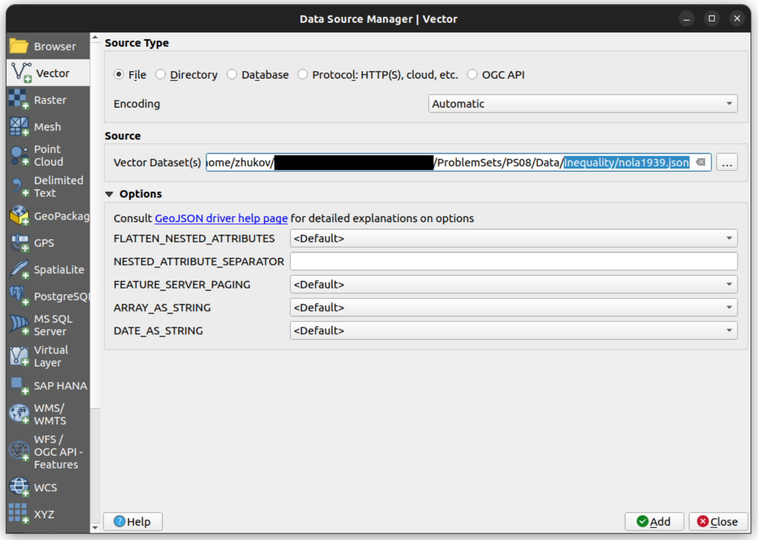
You can adjust the symbology to see how the `flood_mean` variable is distributed



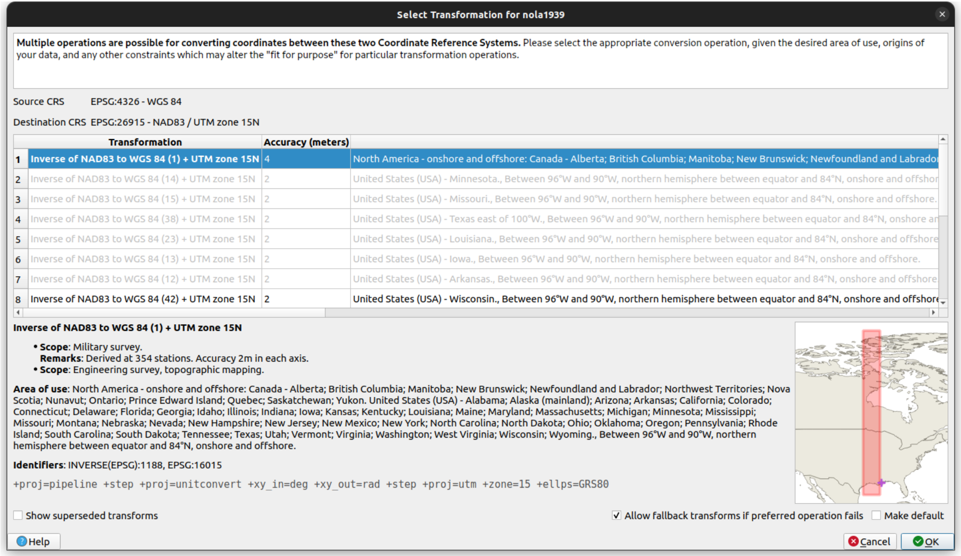
Some of the worst-affected areas seem to be in the city's north (Lakeview, Gentilly), southeast (lower 9th Ward) and wetlands to the east



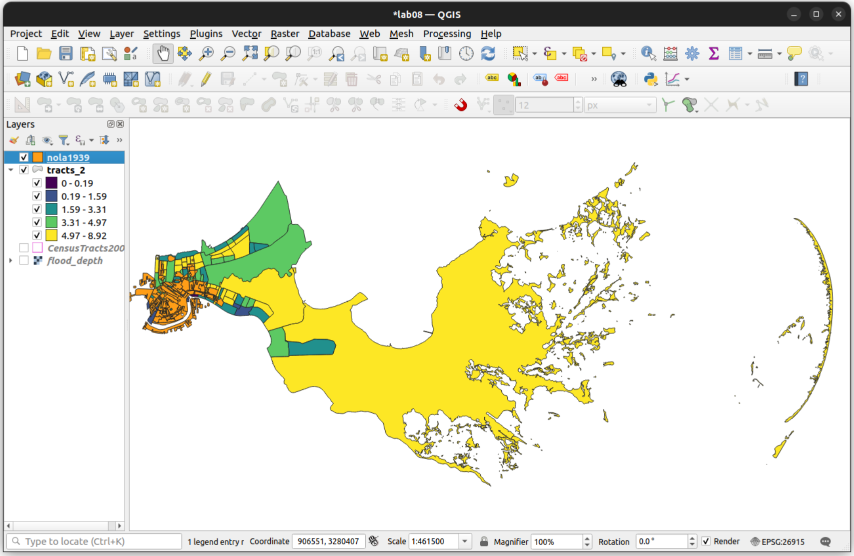
Let's now see if historically "redlined" areas were disproportionately affected. Load the `nola1939.geojson` file from `Data/Inequality/`



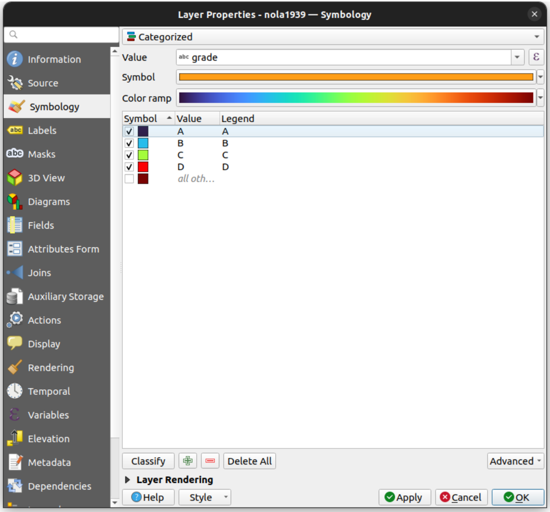
QGIS may prompt you to select a transformation method (to reproject this layer from WGS84 to UTM). Pick the top-listed one and click `OK`



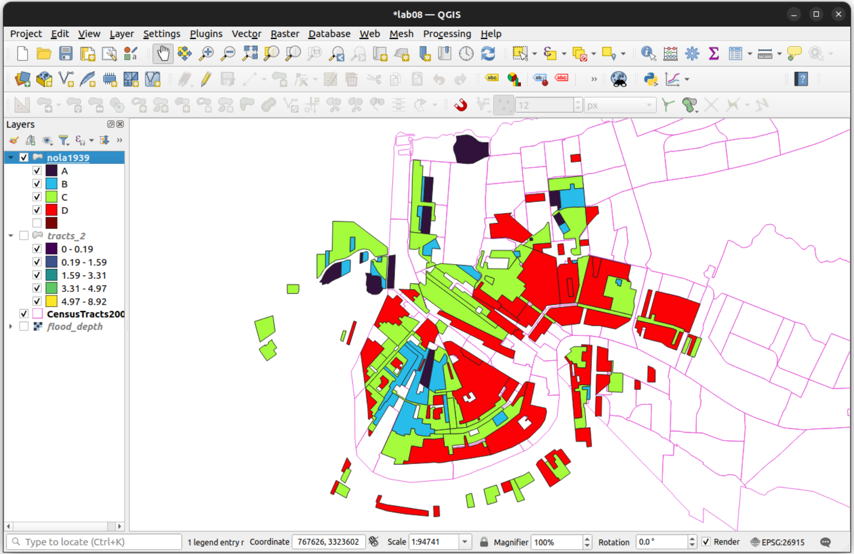
If all goes well, the `nola1939` layer should appear on the map



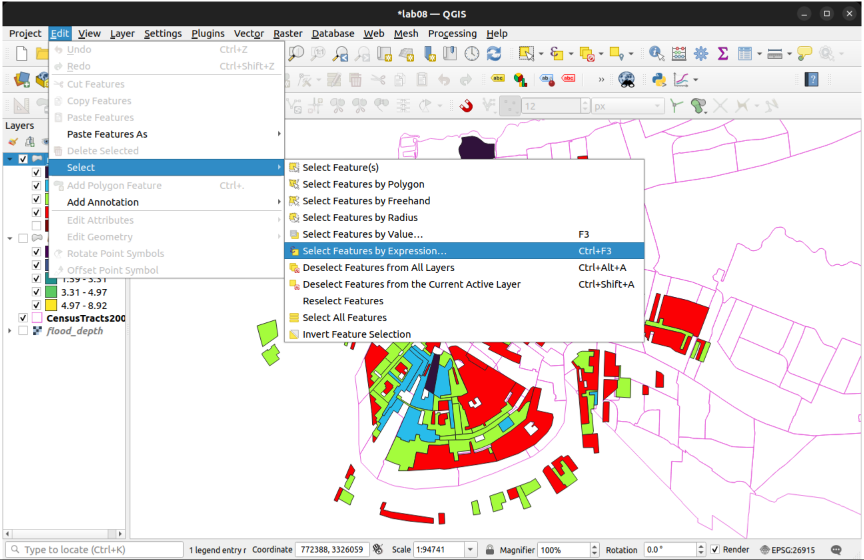
Let's color these polygons by `grade` (Categorized). Grade `D` represents the redlined areas that the HOLC considered too "hazardous" for home loans



These grades made it difficult or impossible for people to access mortgage financing and become homeowners. The brunt of redlining fell on neighborhoods of color.

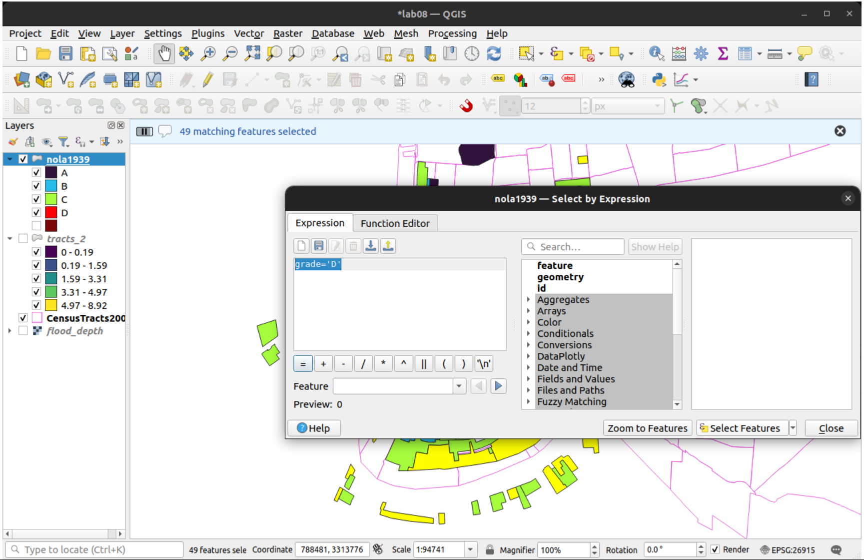


Let's separate out the areas with a grade of D. Go to Edit menu → Select → Select Features by Expression...

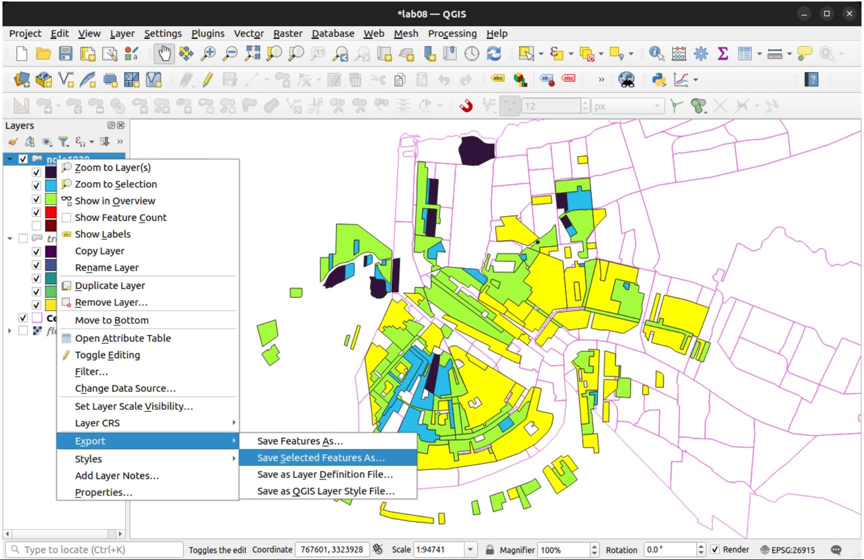


On the next screen, set `Expression` to `grade = 'D'`

Click `Select Features` and close this window



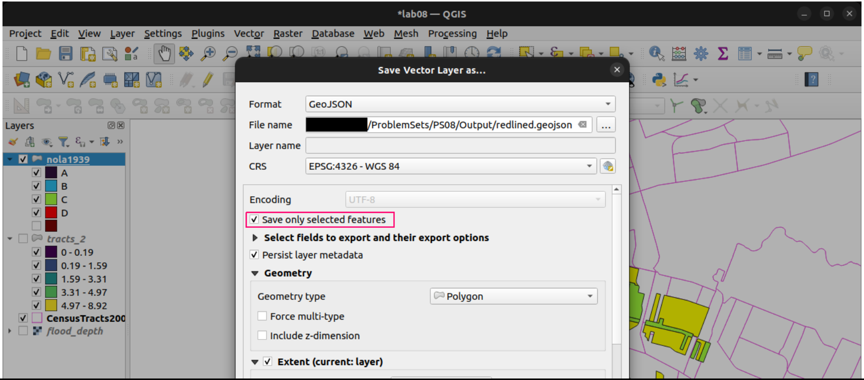
Let's export the selected features to a new file. Right-click on `nola1939` in the layer menu, then `Export` → `Save Selected Features As...`



On the next screen set

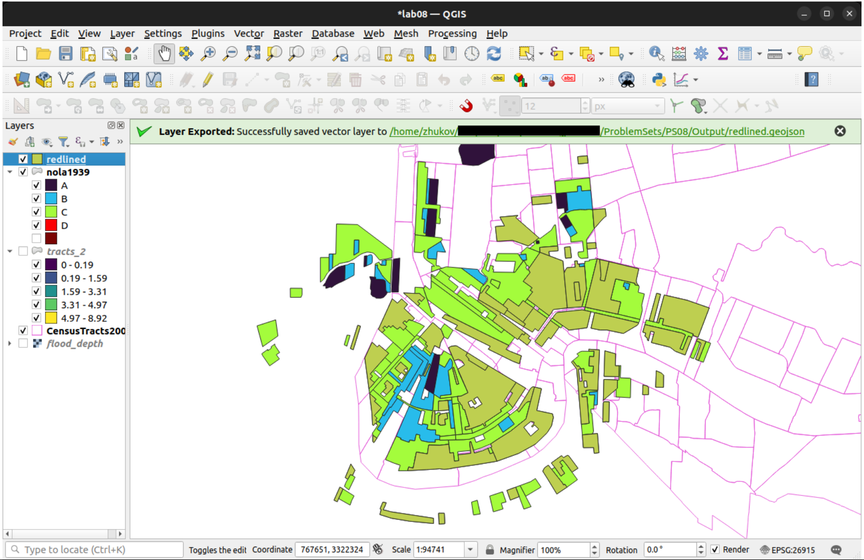
- Format = GeoJSON
- File name = redlined.geojson
- ✓ Save only selected features
- Geometry type = Polygon

Click OK

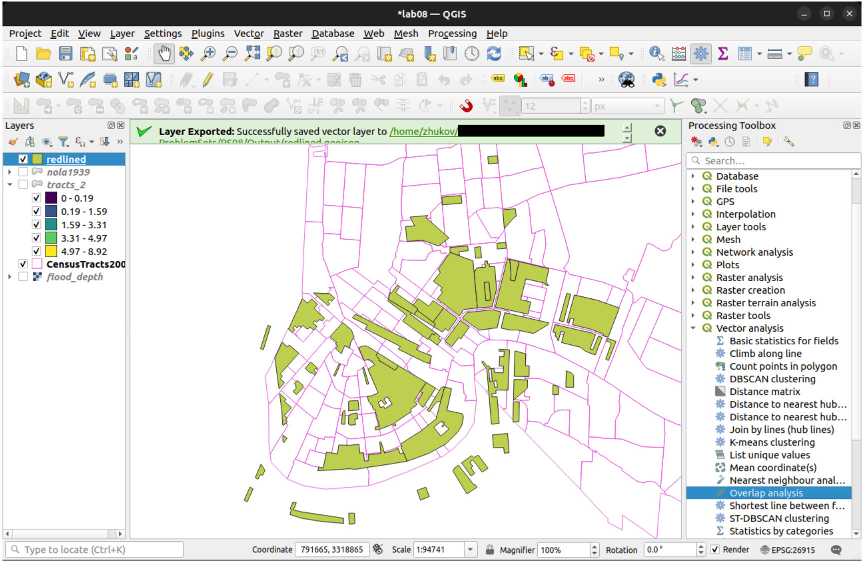


The `redlined` layer should appear in your project window.

Let's calculate the proportion of each census block that was redlined in 1930s



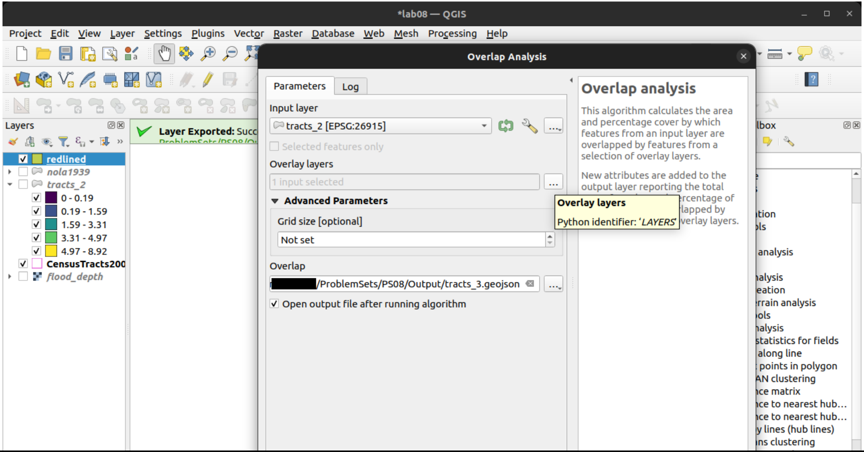
Go to Processing Toolbox, then Vector analysis → Overlap analysis



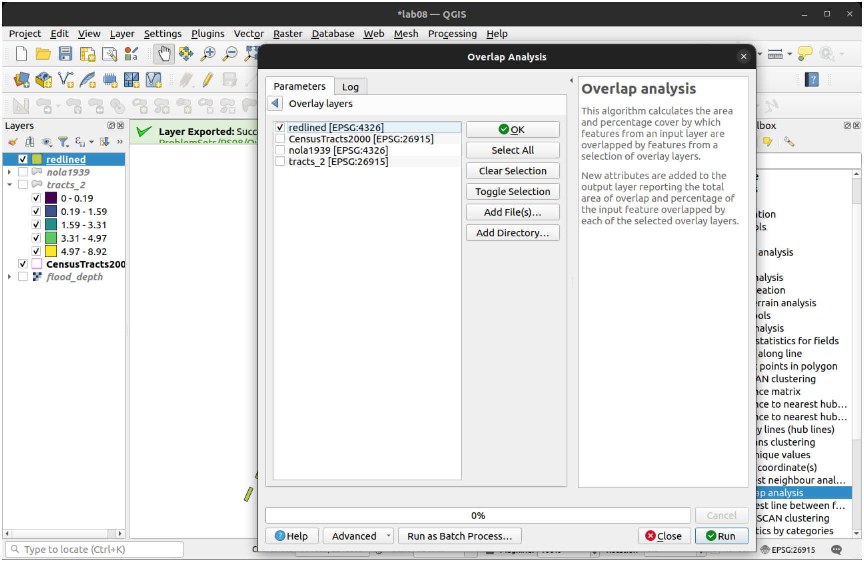
In the Overlap Analysis window, set

- `Input layer` = `tracts_2`
- `Overlap` (save to file) = `tracts_3.geojson`

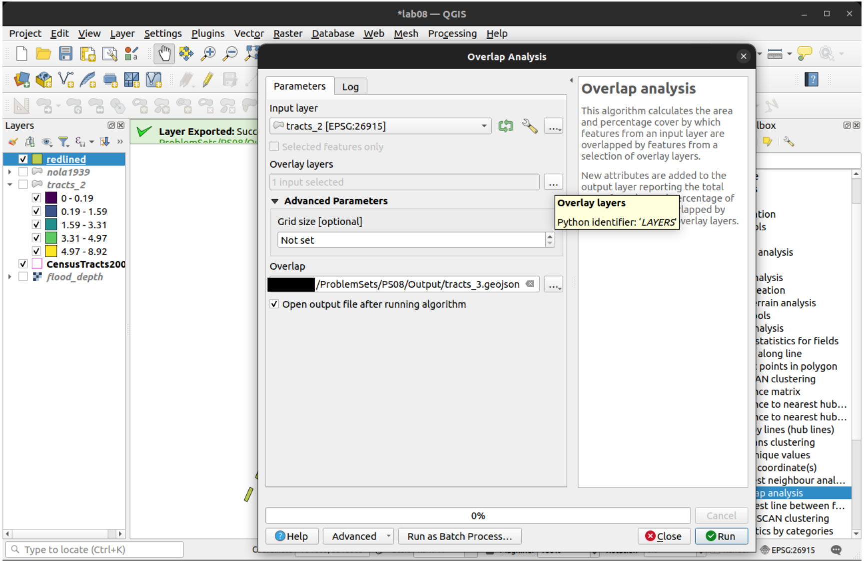
Click on the `[...]` button next to `Overlay layers`



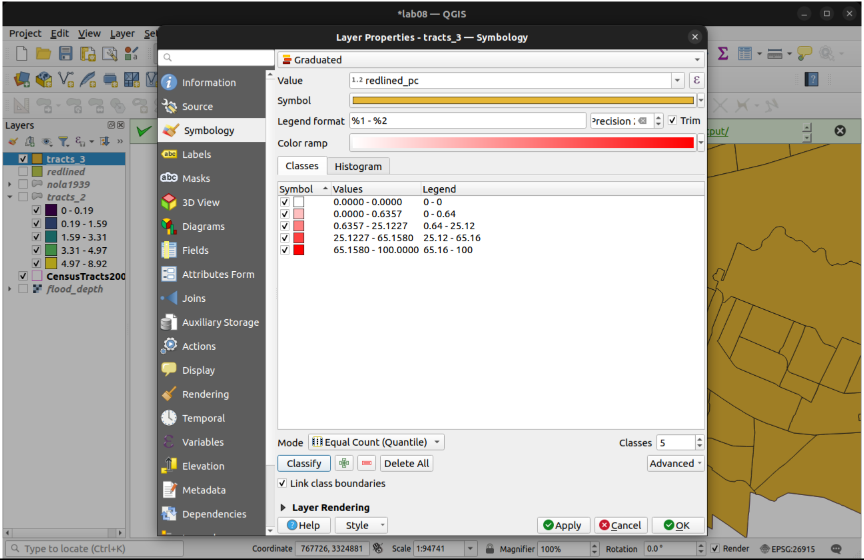
Select ✓ redlined. Click OK



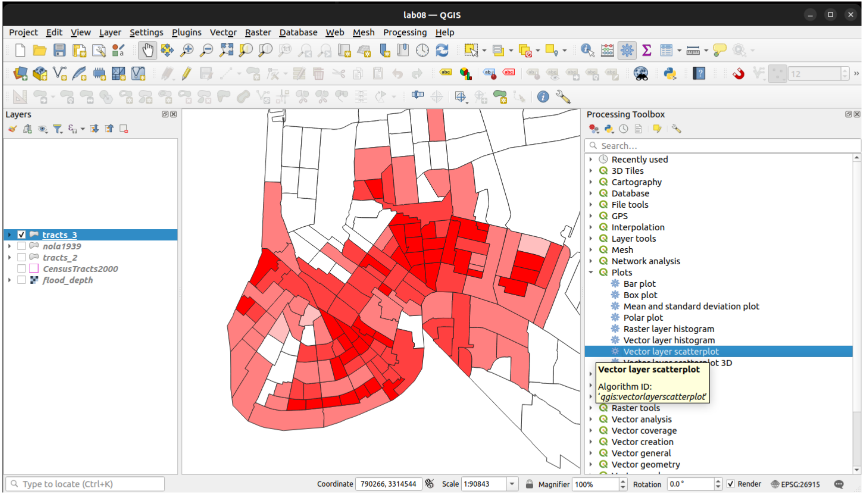
Click `Run`. This will add a new layer, `tracts_3`, to your project



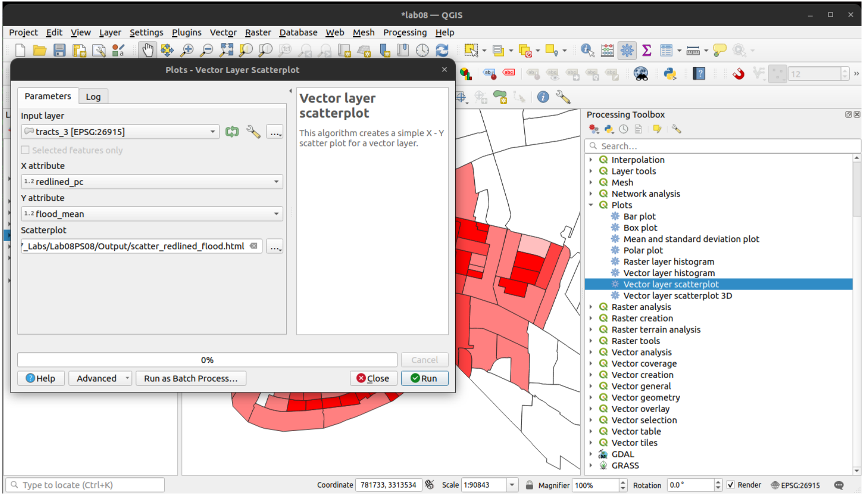
You can adjust the symbology to see how the `redlined_pc` variable (percent of tract with grade D) is distributed



Let's make a quick scatterplot to see if redlined areas saw more flooding. Go to `Processing Toolbox → Plots → Vector layer scatterplot`

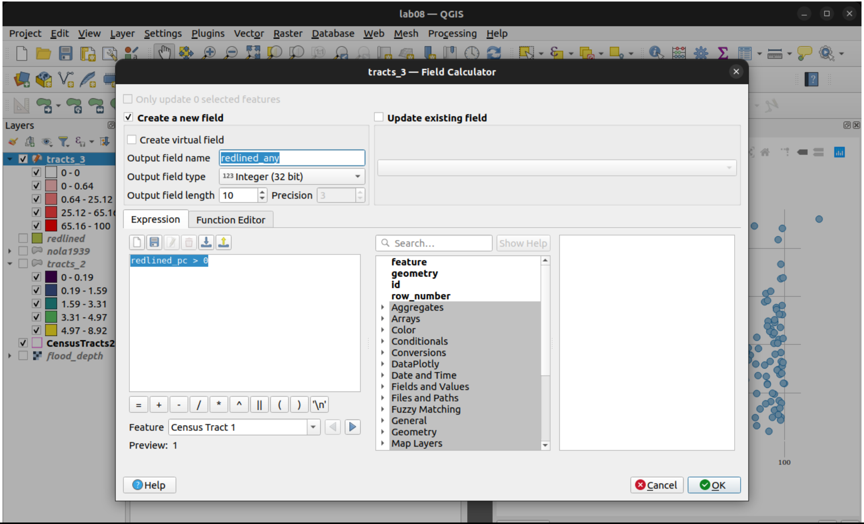


In the next window, set Input layer = tracts_3; X attribute = redlined_pc; Y attribute = flood_mean; Save scatterplot as scatter_redlined_flood.html and open in browser

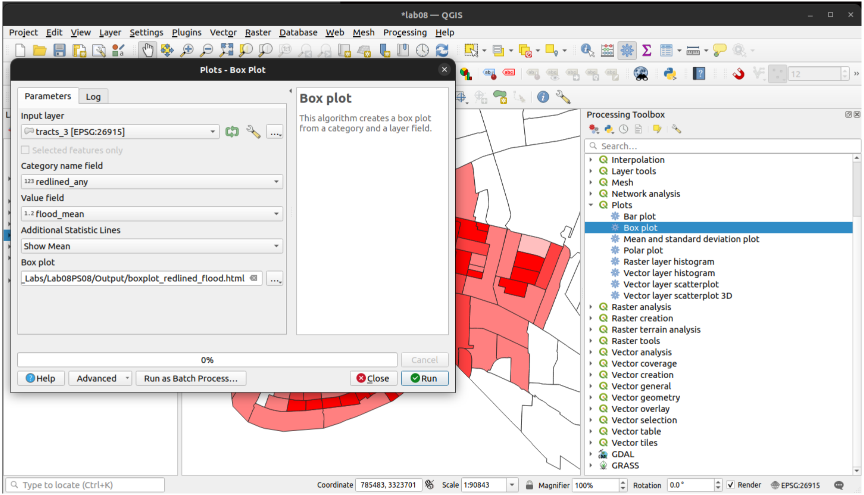


There doesn't seem to be much of a relationship here. Census tracts with more redlined areas did not experience more flooding than less-redlined areas

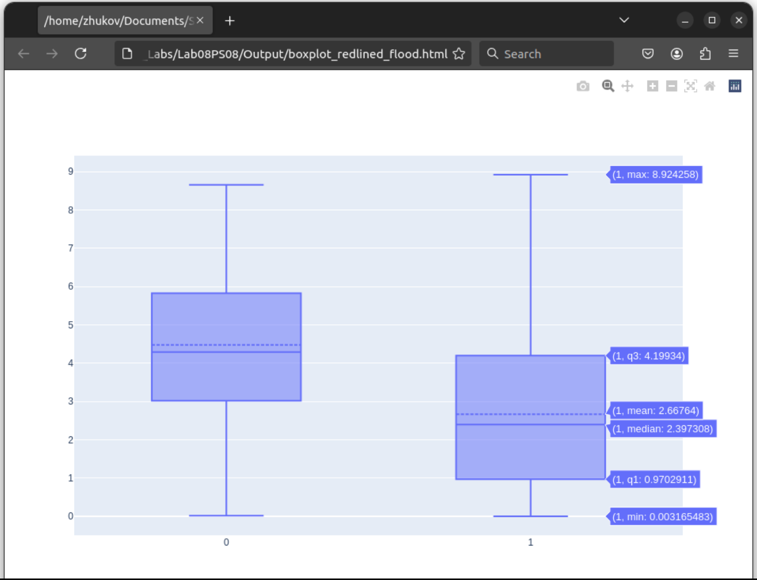
What if we compared tracts with *any* redlining to those with no redlining? Open the Field Calculator for `tracts_3` and set `name: redlined_any`, `type: Integer`, `Expression: redlined_pc > 0`, Click `OK`



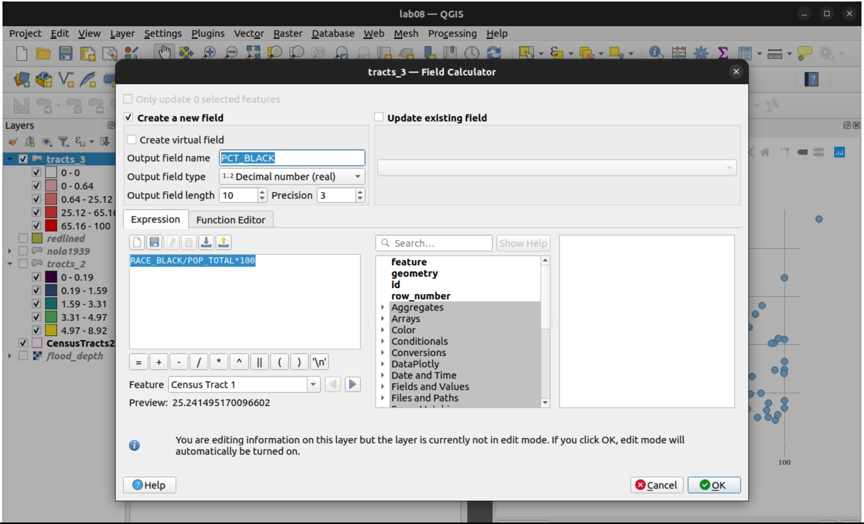
Back in the Geoprocessing Toolbox, select Plots → Box Plot; Input layer = tracts_3; Category name field = redlined_any; value field = flood_mean. Save as boxplot_redlined_flood.html and open in browser



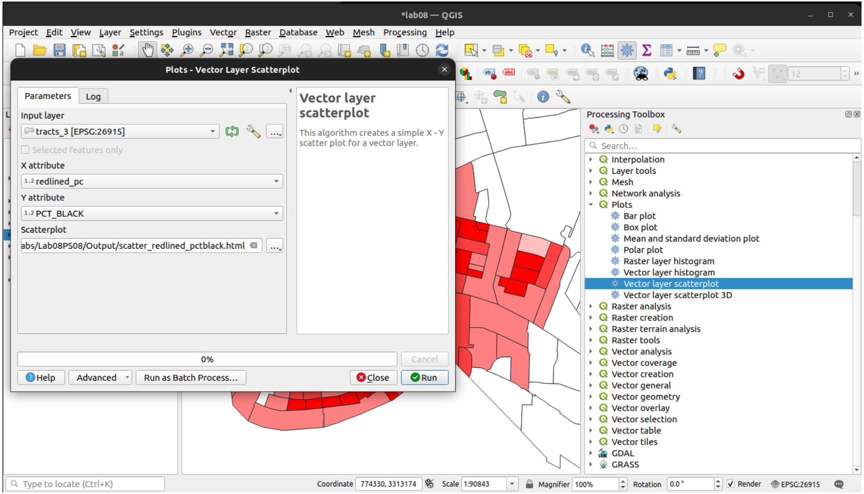
The boxplot suggests that, if anything, redlined areas saw less flooding, on average



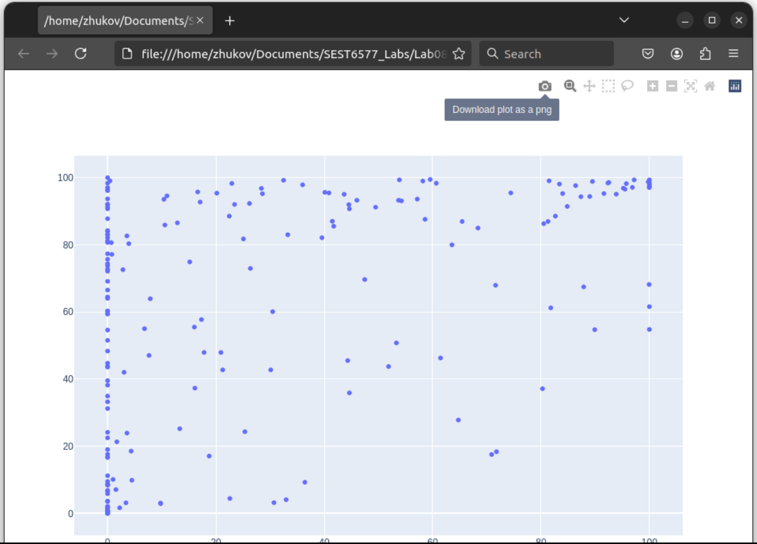
Let's look at redlining's relationship with neighborhood demographics. Go to Field Calculator for `tracts_3` and create a new variable, with `name`: `PCT_BLACK`, `type`: `Decimal number`, `Expression`: `RACE_BLACK / POP_TOTAL * 100`. Click `OK`



Let's make a scatterplot with `redlined_pc` on the X axis and `PCT_BLACK` on the Y axis.

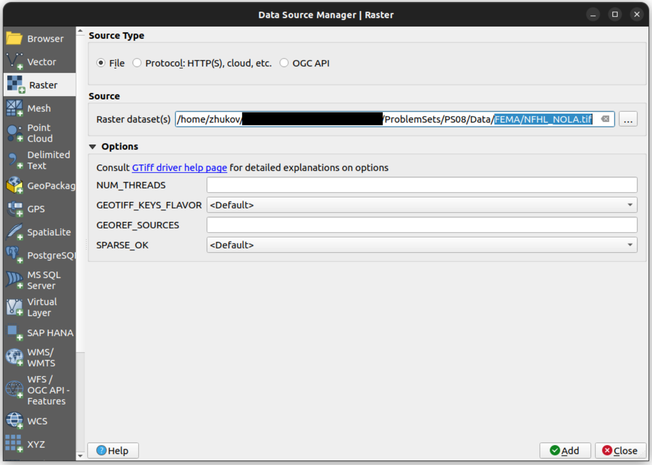


While there is a lot of variation among non-redlined neighborhoods, those closer to 100% redlining still have an overwhelmingly Black population

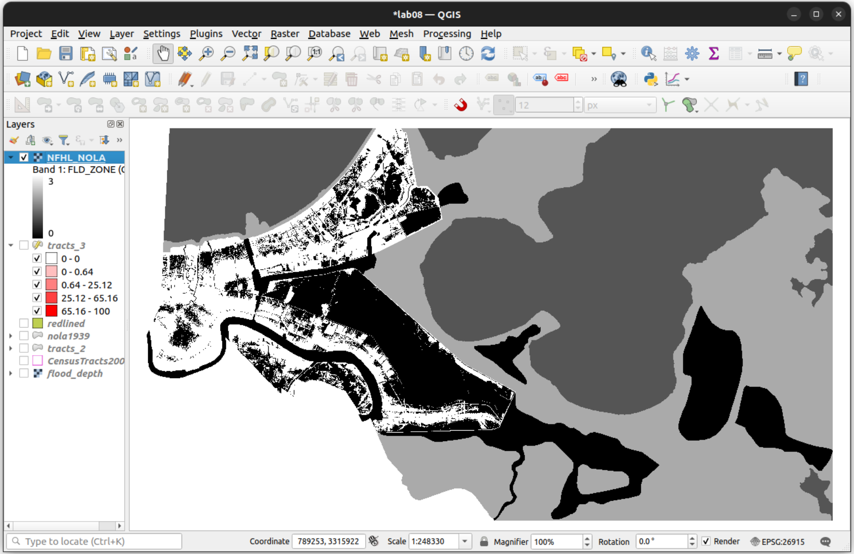


Raster reclassification

Sometimes, we need to do some pre-processing to prepare raster data for analysis. Suppose we want to know the proportion of a census tract classified by FEMA as "high flood risk". Load the raster `NFHL_NOLA.tif` from the folder `Data/FEMA/`



The raster appears to have four unique classes, titled `FLD_ZONE`. They are on a greyscale ramp by default



FEMA uses several designations of flood risk, which are used by flood insurance providers and home lenders

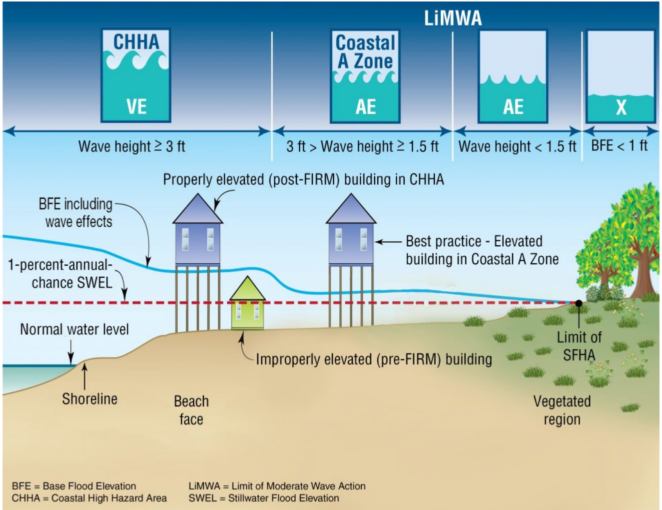
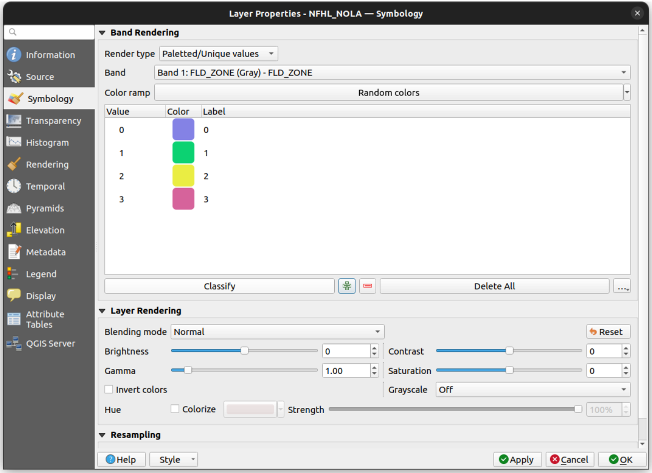
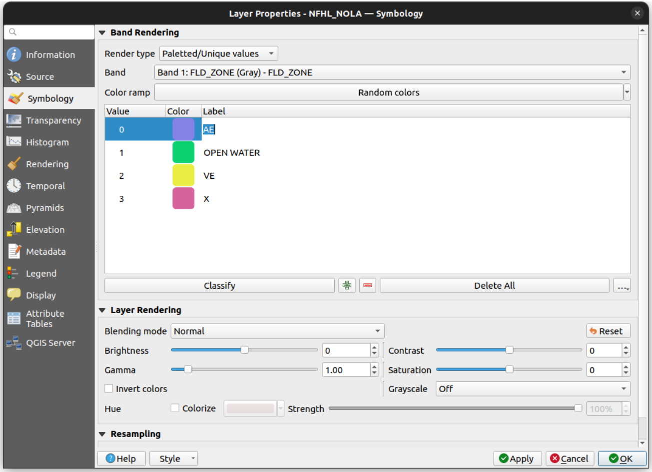


Figure 13: NFHL codes

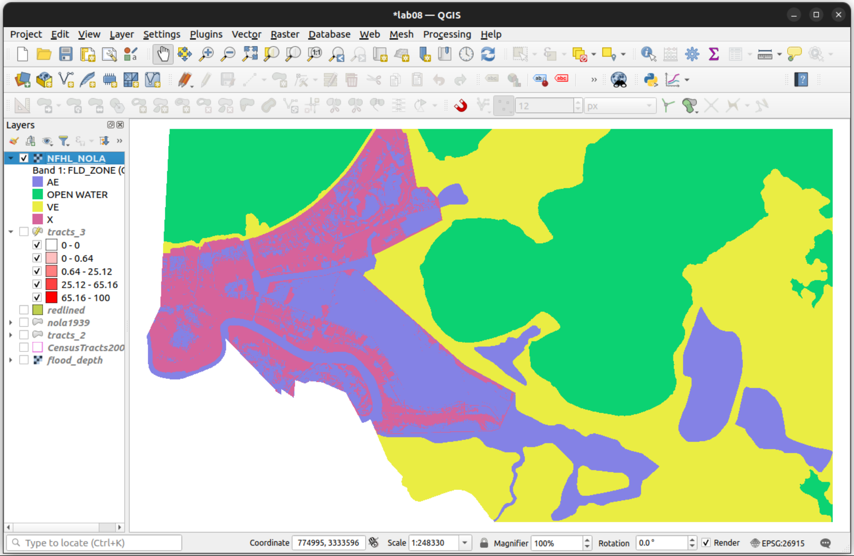
QGIS reads these codes as integers, which correspond to. . .



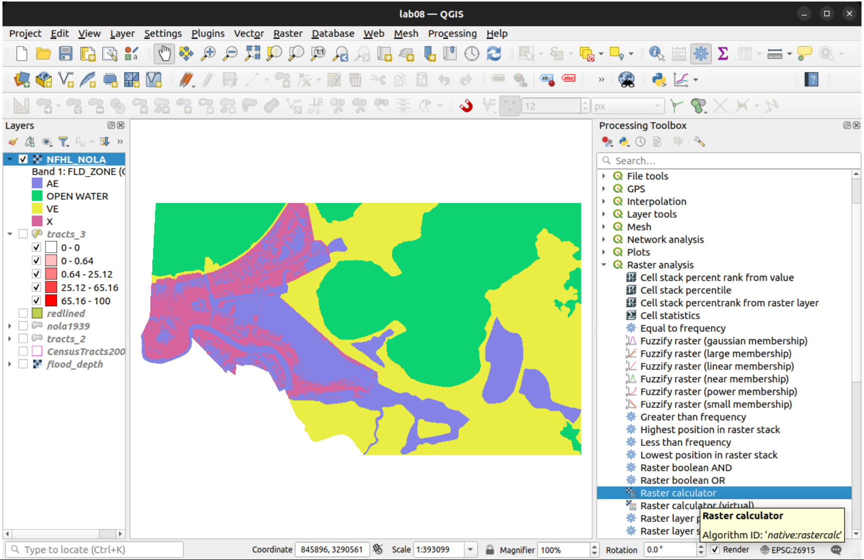
0 → "AE" (high flood risk), 1 → "OPEN WATER",
2 → "VE" (coastal high hazard area), 3 → "X" (moderate-to-low risk)



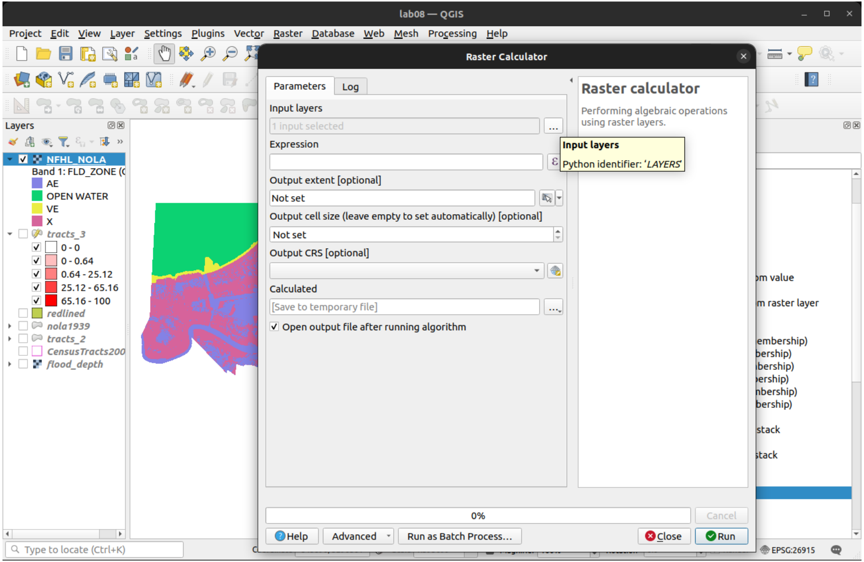
What we want to know is how much of a census tract is covered by "high flood risk" zones (AE, or 0). Let's extract just this part of the raster



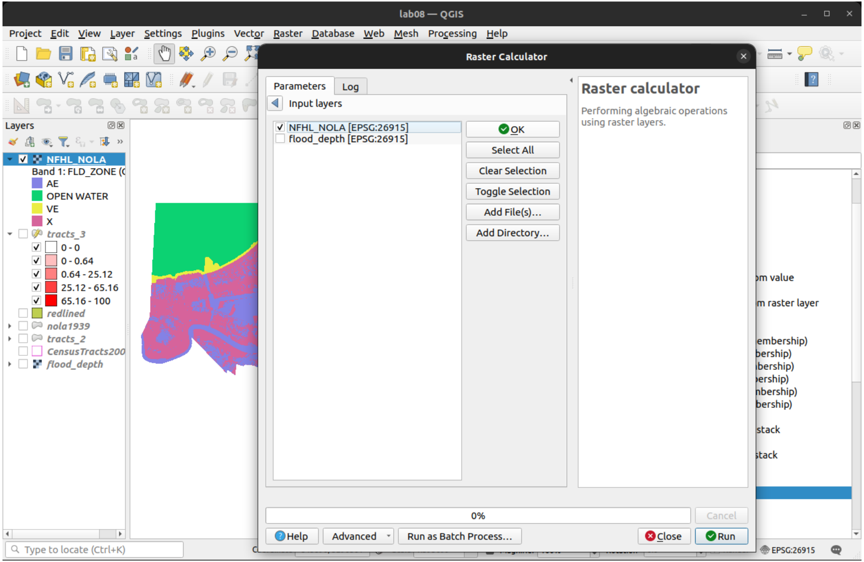
Go back to the Processing Toolbox → Raster analysis → Raster calculator



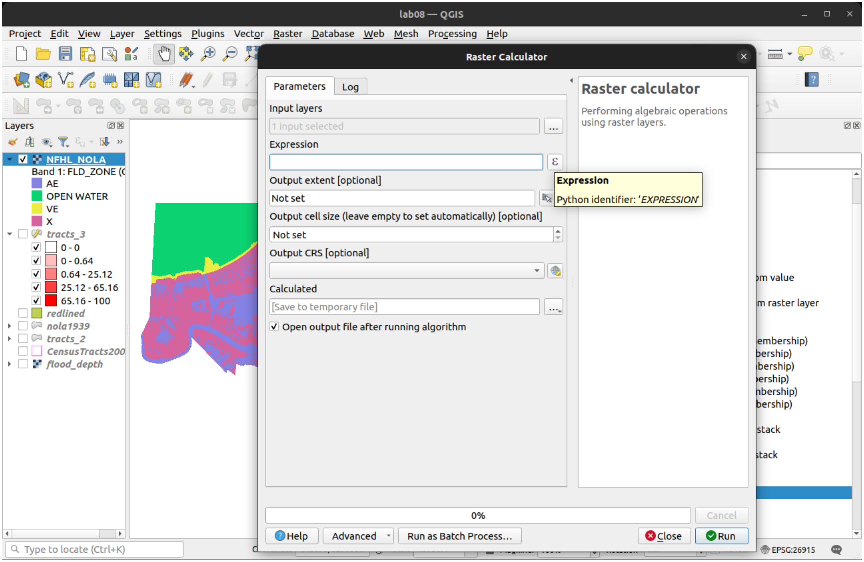
In the Raster Calculator window, click on the [...] box next to `Input layers`



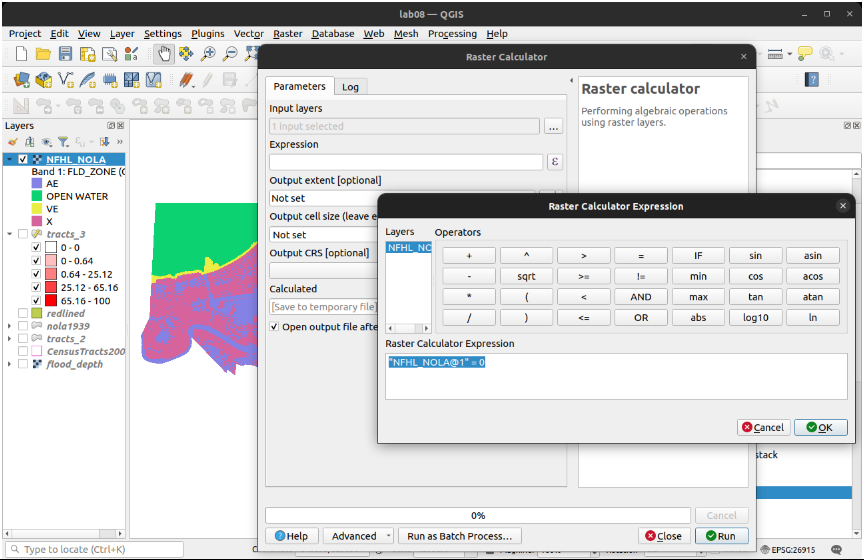
Select ✓ NFHL_NOLA. Click OK



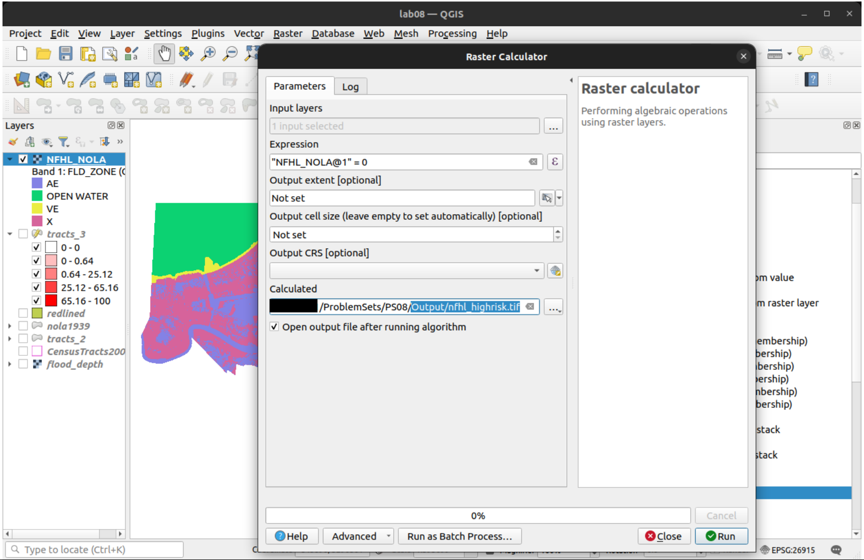
Click on the ϵ button next to `Expression`



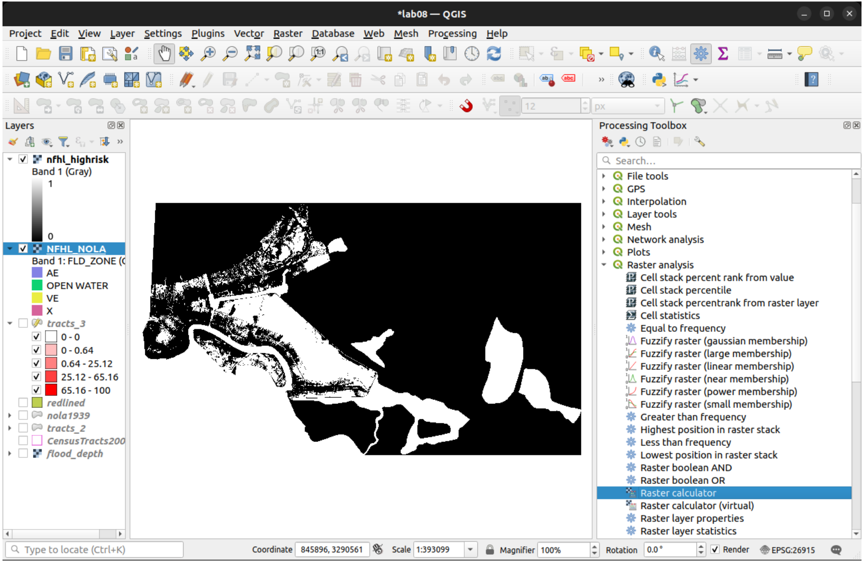
For the Expression, enter "NFHL_NOLA@1" = 0 (syntax is "layer_name@band_number"; make sure to include the quotation marks). Click OK



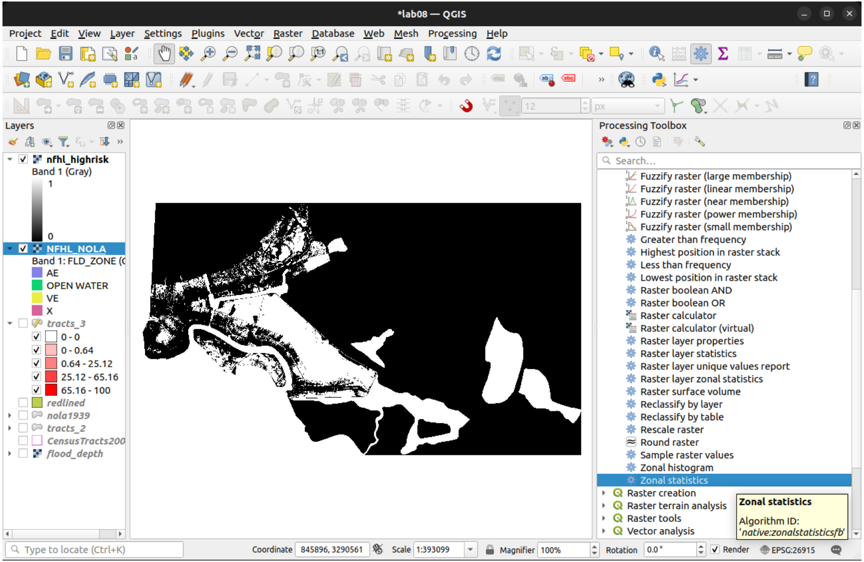
Leave the `Output extent`, `cell size` and `CRS` fields blank (accept defaults). For `Calculated`, save to file as `nfhl_highrisk.tif`. Click `Run`



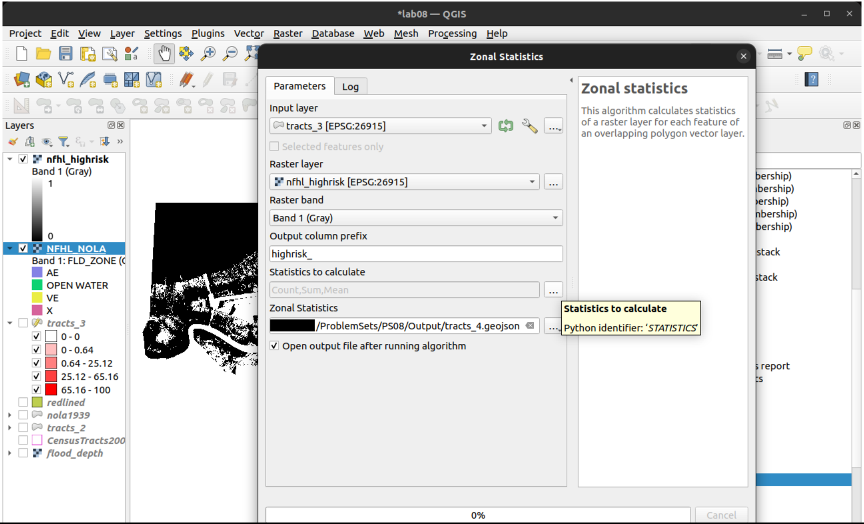
This creates a binary (1/0) raster indicating whether or not pixel is in high-risk zone



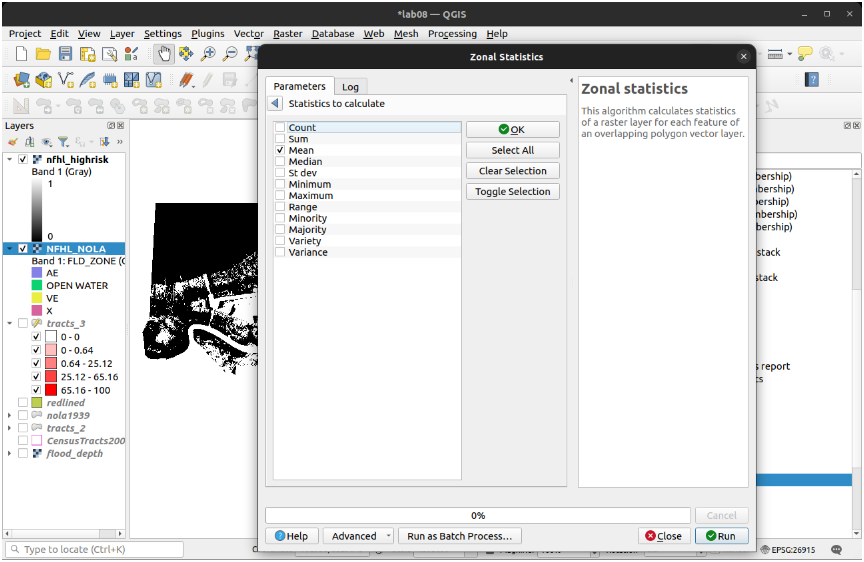
Go back to `Zonal statistics` in the Processing Toolbox



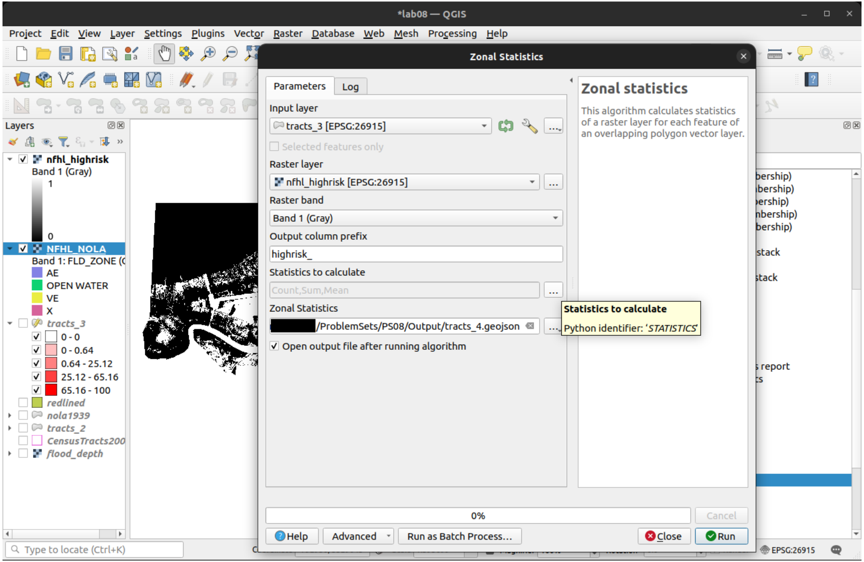
Select tracts_3 as the Input layer, nfhl_highrisk as the Raster layer, set the column prefix to highrisk_. CLick on the [...] button next to Statistics to calculate



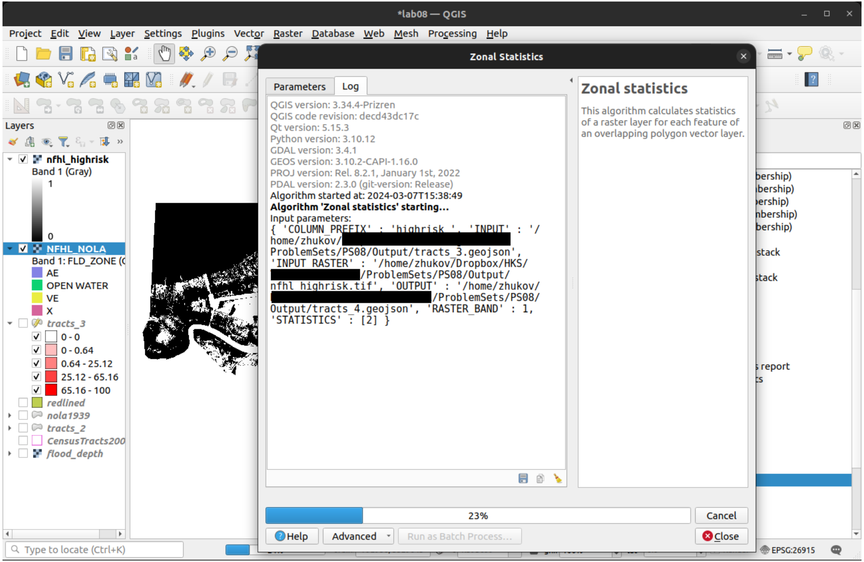
Select ✓ Mean. Click OK



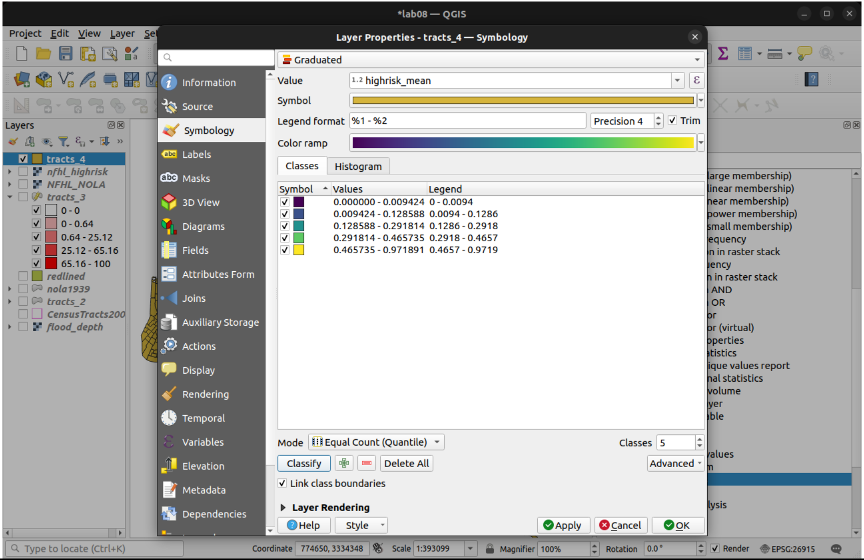
Save the output to a file named `tracts_4.geojson`. Click `Run`



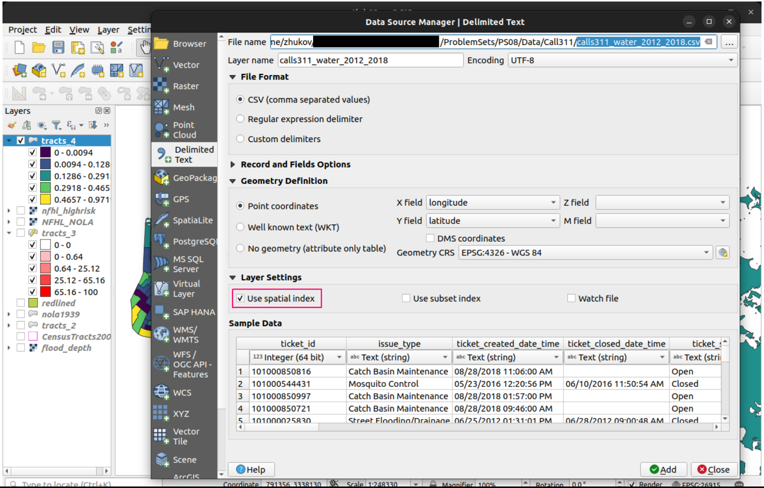
This may take a few minutes. Be strong. Don't give up



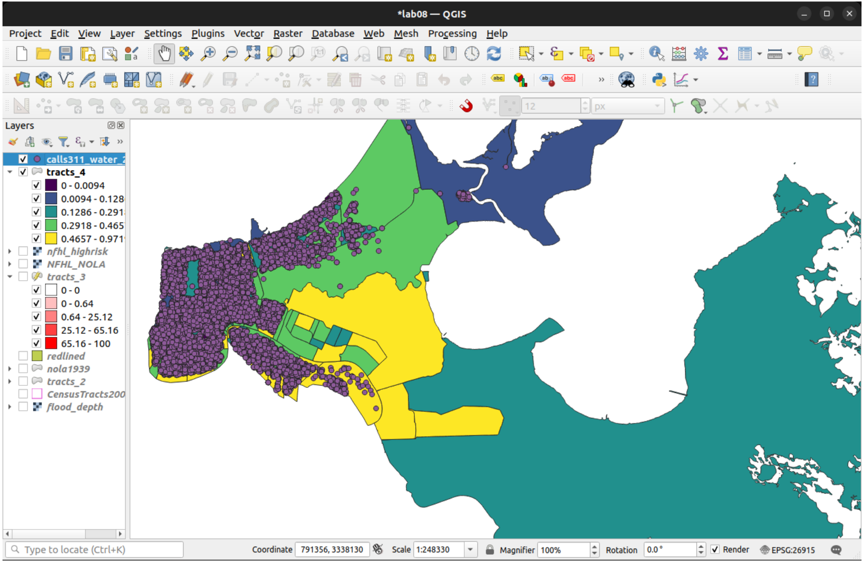
When the `tracts_4` layer finally loads, you can change the symbology and explore the distribution of the `highrisk_mean` variable



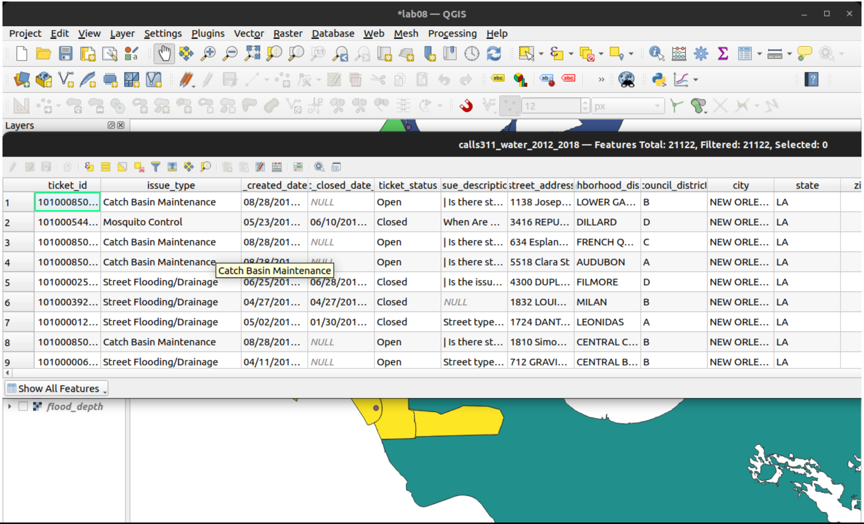
The final ingredient we will need is the 311 call data. Load the delimited text file `calls311_water_2012_2018.csv` from the `Data/Call311/` folder. Set the `Geometry Definition` to `Point coordinates` (as shown here), and check the box next to ✓ `Use spatial index` in `Layer Settings`. Click `Add`



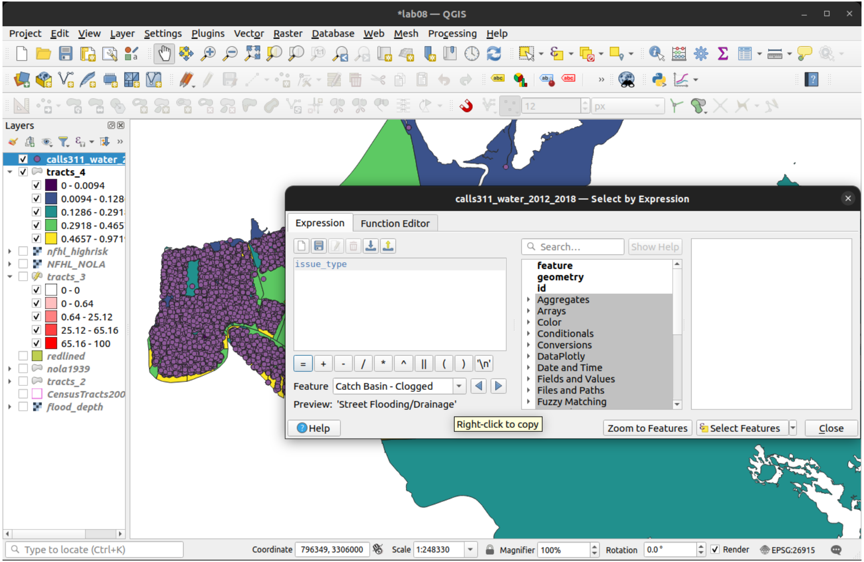
Note that the 311 calls are available only for New Orleans, not St. Bernard Parish



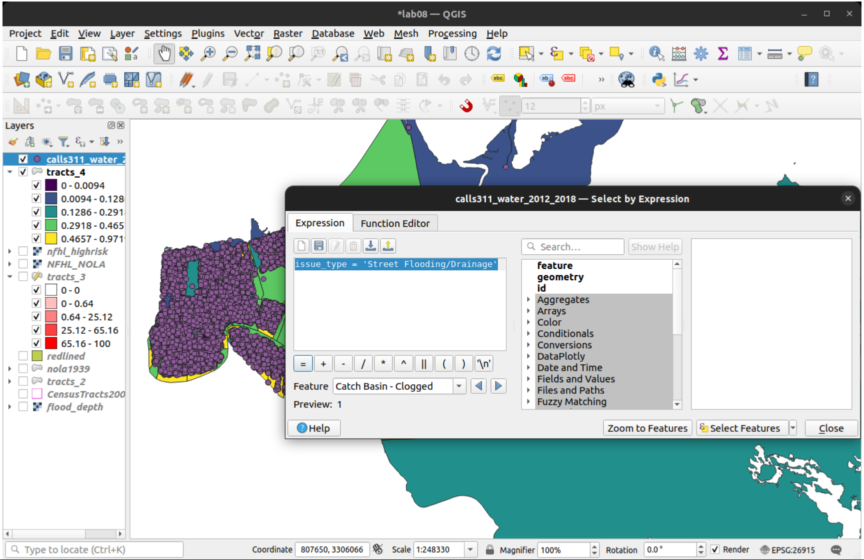
Open the Attribute Table for the `calls311_*` layer, and look at the `issue_type` field. There are three types here: "Catch Basin Maintenance", "Mosquito Control", and "Street Flooding/Drainage". Let's create point counts for each of these



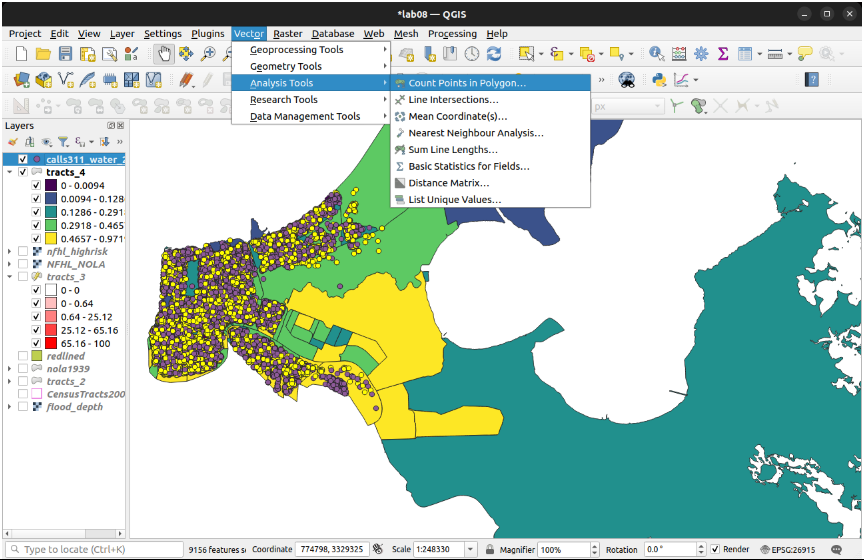
Go to Edit menu → Select → Select Features by Expression...



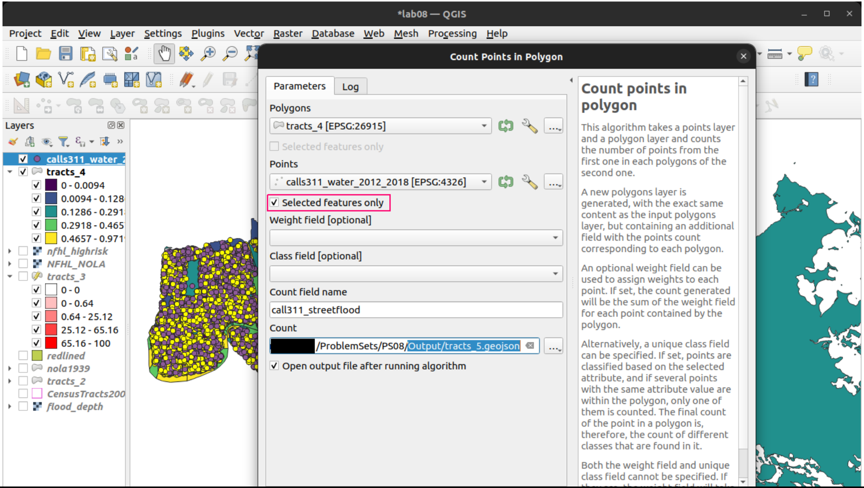
For `Expression`, enter `issue_type = 'Street Flooding/Drainage'` (with single quotation marks). Click `Select Features`



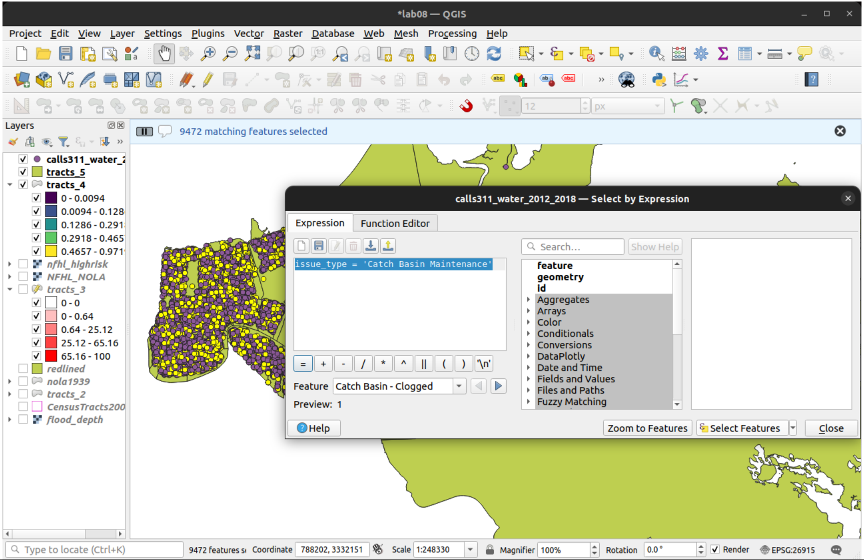
With the points selected, open the `Count Points in Polygon` tool in `Vector` → `Analysis Tools`



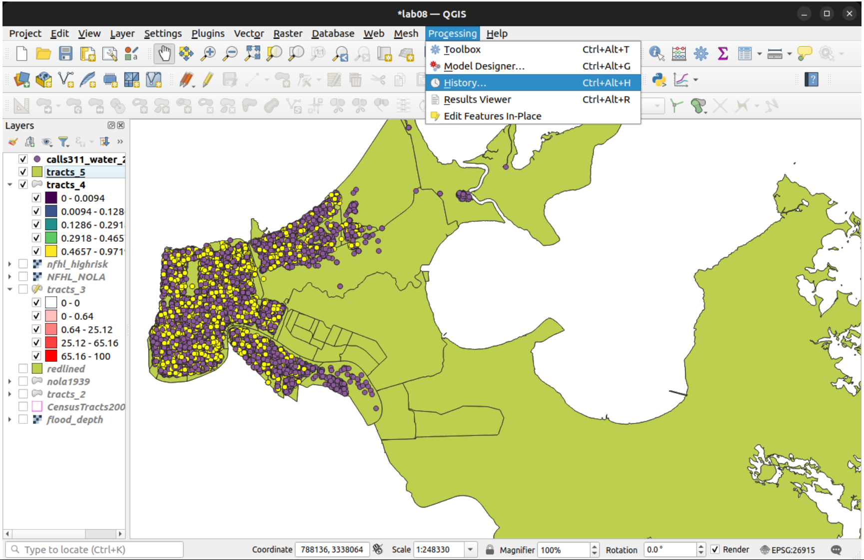
Select `tracts_4` as the Polygons layer and `calls311_*` as the Points layer. Make sure the box next to ✓ `Selected features only` is checked. Name the count field `calls311_streetflood` and save the output to a new file called `tracts_5.geojson`. Click `Run`



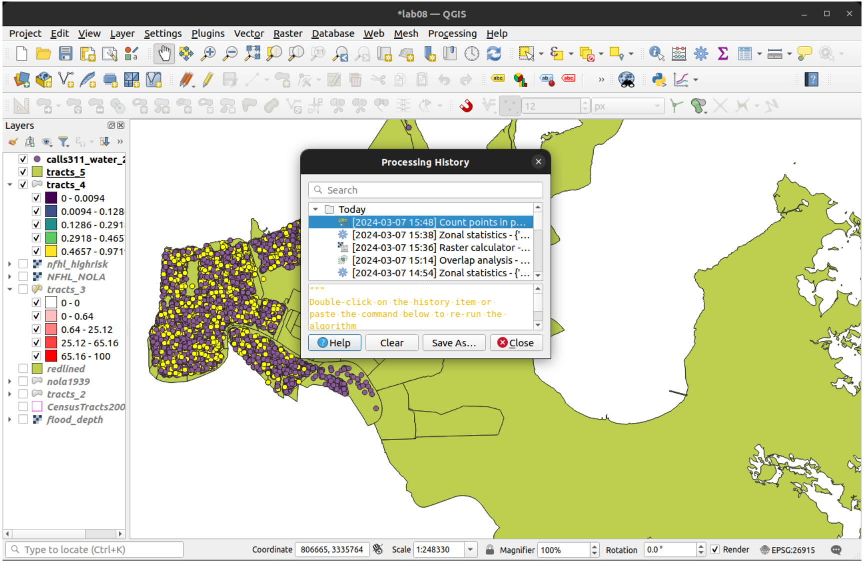
Repeat this process for the other two categories of calls, starting with "Catch Basin Maintenance"



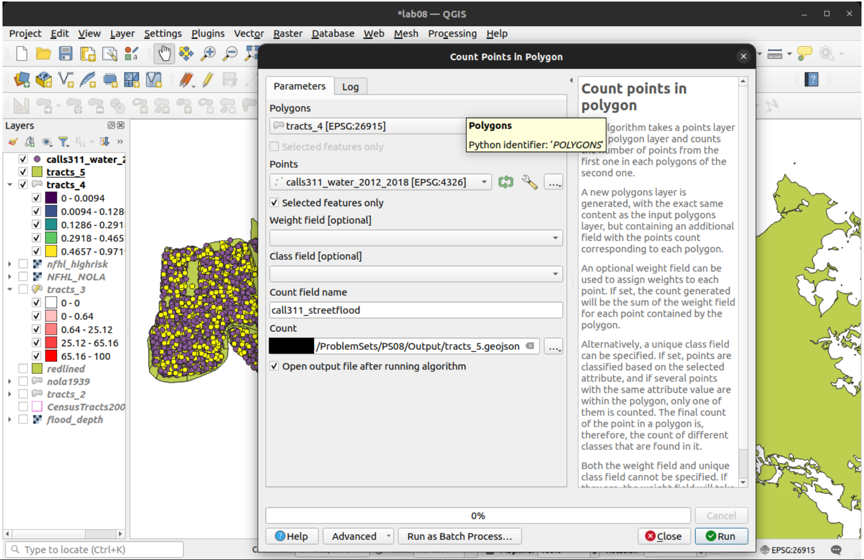
To save a bit of time, you can call up the most recently-used geoprocessing options by going to `Processing` menu → `History...`



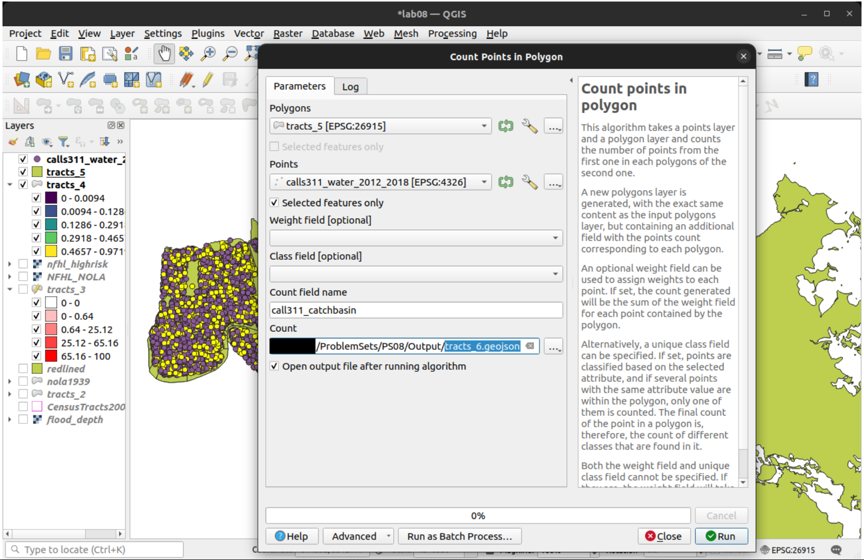
Double-click on the Count points in polygon operation



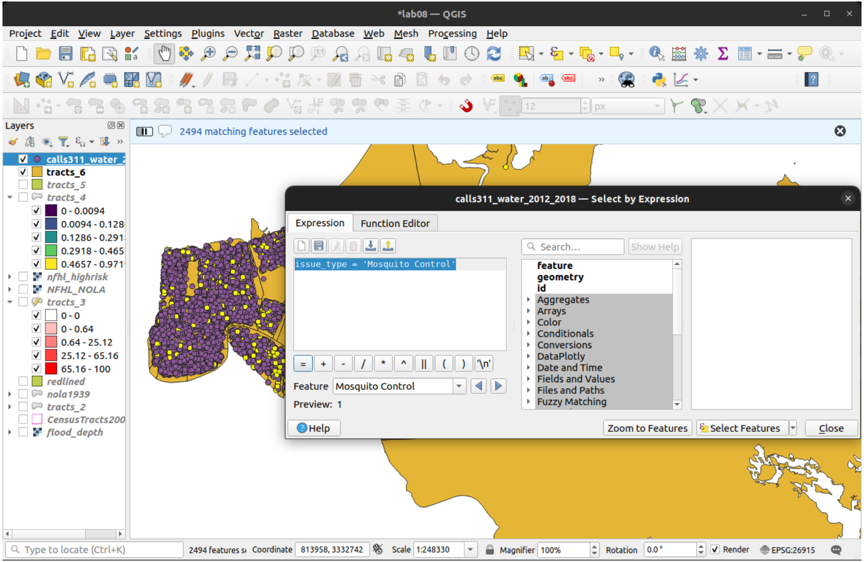
This will restore the settings you just used to create `tracts_4`. Now we just need to update the parameters



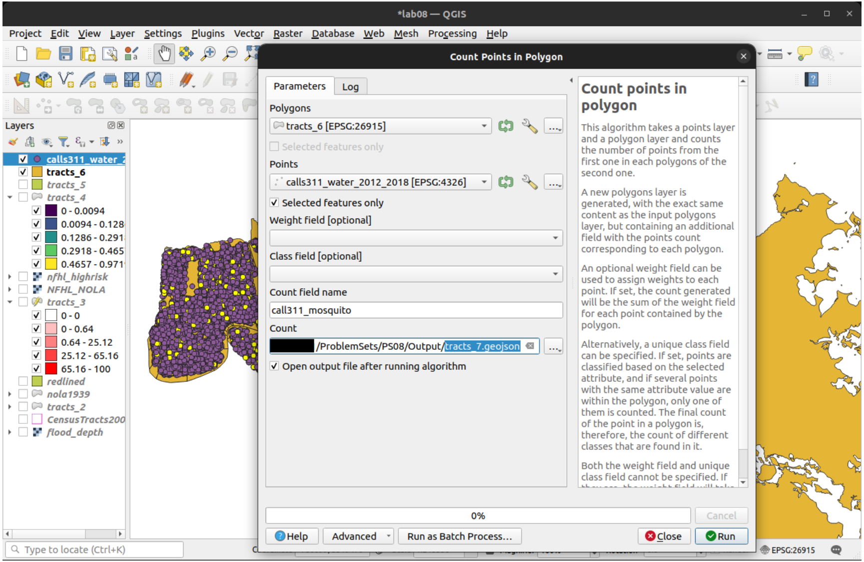
Change Polygons to tracts_5, change field name to calls311_catchbasin, and save the file as tracts_6.geojson. Click Run



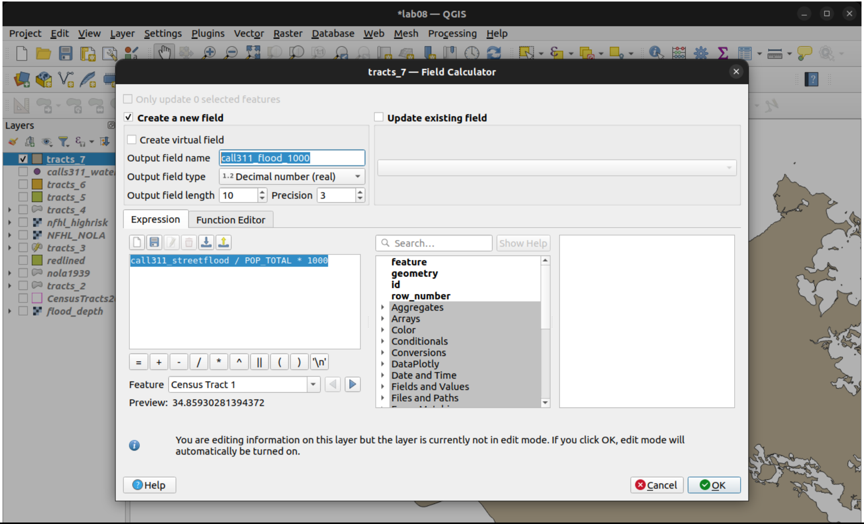
And one more time with `issue_type = 'Mosquito Control'`



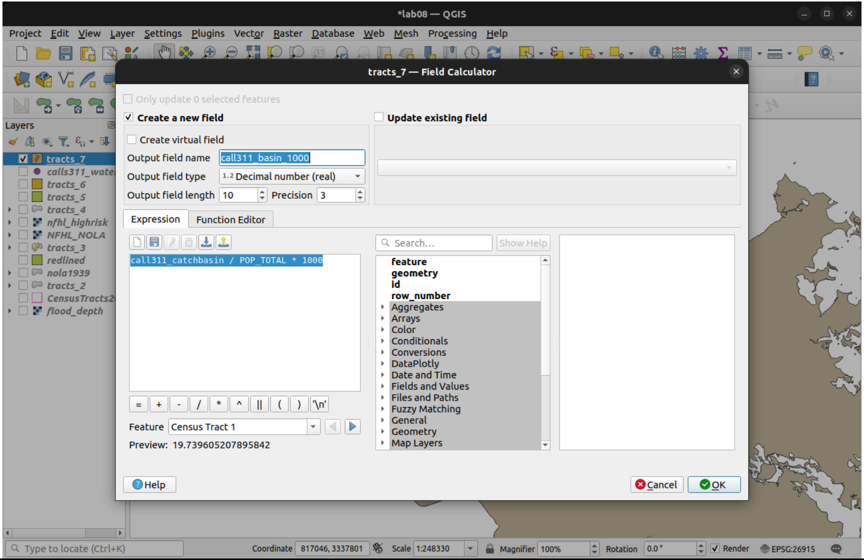
Set `Polygons` to `tracts_6`, field name to `calls311_mosquito`, save as `tracts_7`



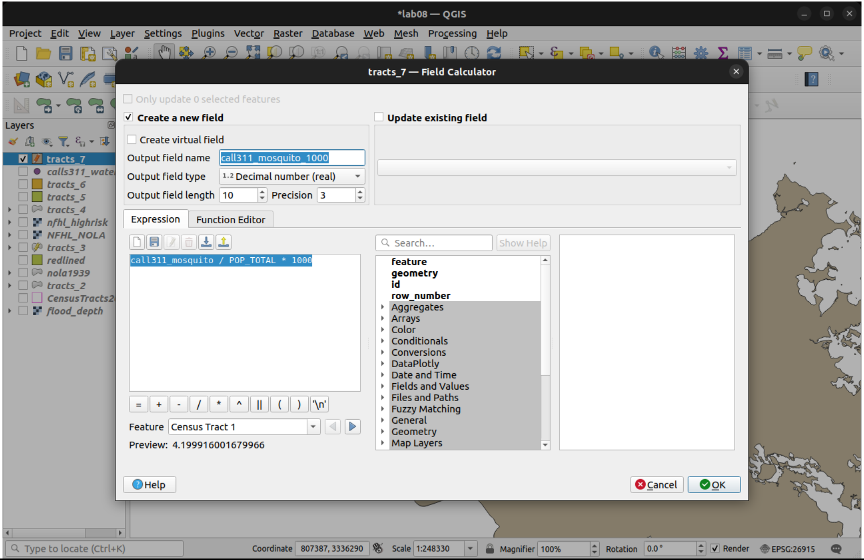
Let's make some per capita measures. Open the Field Calculator for `tracts_7`, create a new field with name `calls311_flood_1000` of type `Decimal number`. Set Expression to `calls311_streetflood / POP_TOTAL * 1000`



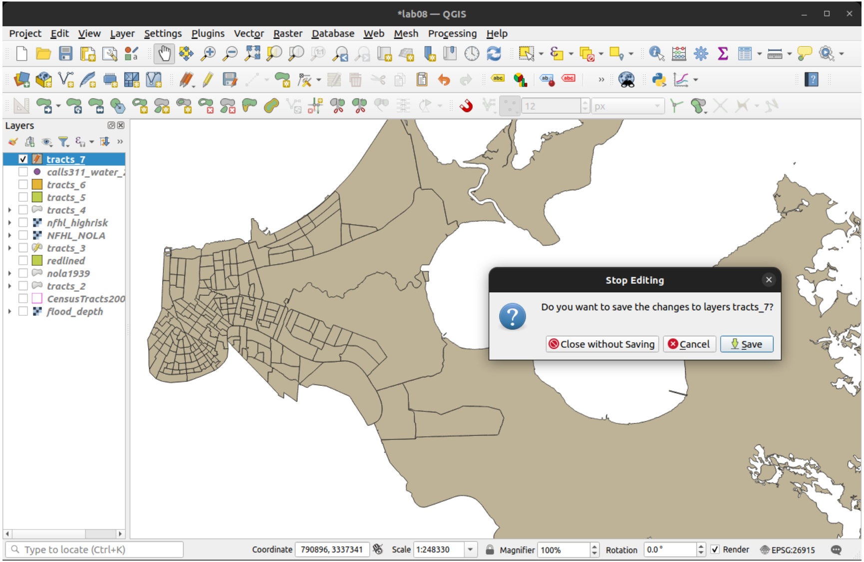
Also create a new field named `calls311_basin_1000` of type `Decimal number`.
Set Expression to `calls311_catchbasin / POP_TOTAL * 1000`



And finally, a new field with name `calls311_mosquito_1000` of type `Decimal number`. Set Expression to `calls311_mosquito / POP_TOTAL * 1000`



Now is a good time to save your progress, to `tracts_7` and to the project



Problem Set 8

Your assignment (if using QGIS): create two scatterplots

1. Flood risk and Katrina flood depth
   - `highrisk_mean` on x-axis
   - `flood_mean` on y-axis
   - name the file `nfhl_katrina.png`
2. Flood risk and per capita 311 calls about street flooding
   - `highrisk_mean` on x-axis
   - `calls311_flood_1000` on y-axis
   - name the file `nfhl_311flood.png`

- upload both plots to Canvas

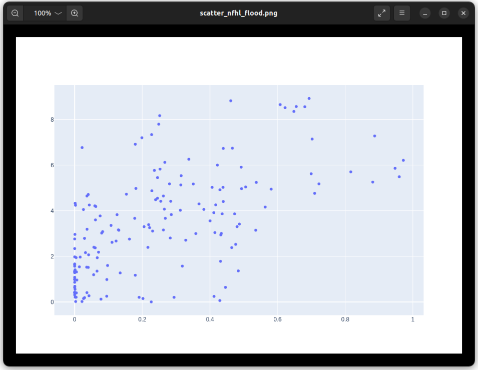


Figure 14: Can you make this?

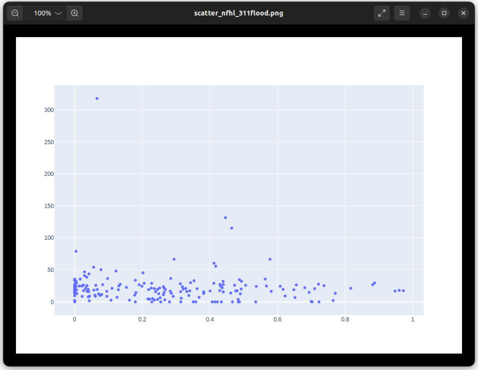


Figure 15: And this?

R

Loading R packages

To implement these steps in R, we will be using the `sf`, `terra` and `ggplot2` packages

```
library(sf)
library(terra)
library(ggplot2)
```

NOTE: The demo code for R is in `ps08_demo.R` on RStudio Cloud, and in `PS08.zip` (posted on Canvas).

Zonal statistics

Let's load the *census tracts data* into R, using `sf::read_sf()`:

```
tracts2000 <- sf::read_sf("Data/Census/CensusTracts2000.geojson")
```

Check the coordinate reference system of these data

```
sf::st_crs(tracts2000)
```

```
## $input
## [1] "NAD83 / UTM zone 15N"
```

Load the *Katrina flood depth data* into R, using `terra::rast()`:

```
flood_depth <- terra::rast("Data/Katrina/flood_depth.tif")
```

Is the coordinate reference system the same as for `tracts2000`?

```
sf::st_crs(flood_depth)==sf::st_crs(tracts2000)
```

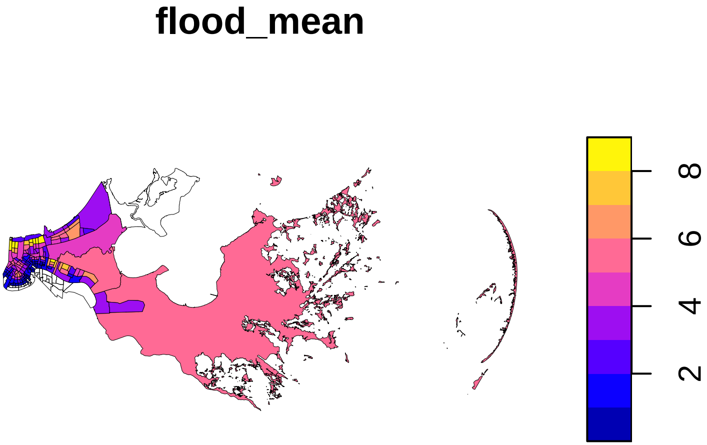
```
## [1] TRUE
```

Yay! That means we can move forward with zonal statistics.

Calculate *zonal statistics*: mean and maximum flood depth in each tract

```
tracts2000$flood_mean <- terra::zonal(x=flood_depth,
  z=terra::vect(tracts2000), fun=mean, na.rm=TRUE)[,1]
tracts2000$flood_max <- terra::zonal(x=flood_depth,
  z=terra::vect(tracts2000), fun=max, na.rm=TRUE)[,1]
```

Plot the results. Note that, unlike QGIS, R assigns `NA` values (instead of 0) to polygons that do not overlap with the raster layer.



Let's load the *HOLC redlining data* into R, using `sf::read_sf()`:

```
nola1939 <- sf::read_sf("Data/Inequality/nola1939.geojson")
```

Is the coordinate reference system the same as for `tracts2000`?

```
sf::st_crs(nola1939)==sf::st_crs(tracts2000)
```

```
## [1] FALSE
```

Uh-oh! Looks like we need to reproject `nola1939`

```
nola1939 <- sf::st_transform(nola1939,crs=sf::st_crs(tracts2000))
sf::st_crs(nola1939)==sf::st_crs(tracts2000)
```

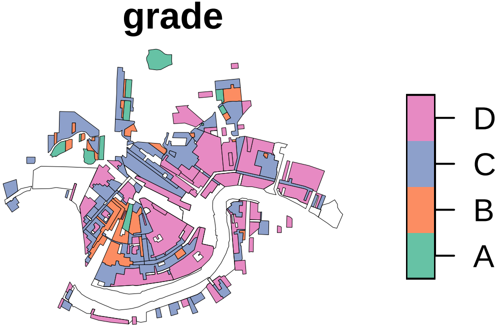
```
## [1] TRUE
```

All clear!

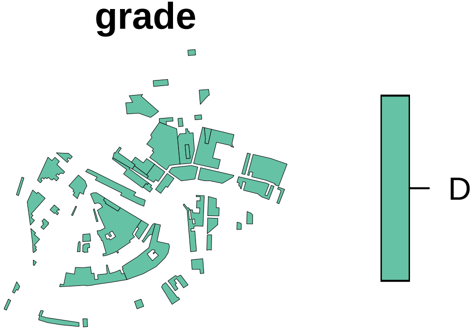
Subset the HOLC data to just "grade D":

```
redlined <- nola1939[nola1939$grade=="D",]
```

```
plot(nola1939["grade"])
```




```
plot(redlined["grade"])
```



Overlap analysis in R takes a few more steps than in QGIS.

Calculate areas of `tracts2000`, and of `tracts2000`+`redlined` intersections:

```
tracts2000$area <- sf::st_area(tracts2000)
tract2red <- sf::st_intersection(tracts2000,sf::st_union(redlined))
tract2red$area_ix <- sf::st_area(tract2red)
```

Aggregate percentage area overlaps by census tract, add this field to `tracts2000`:

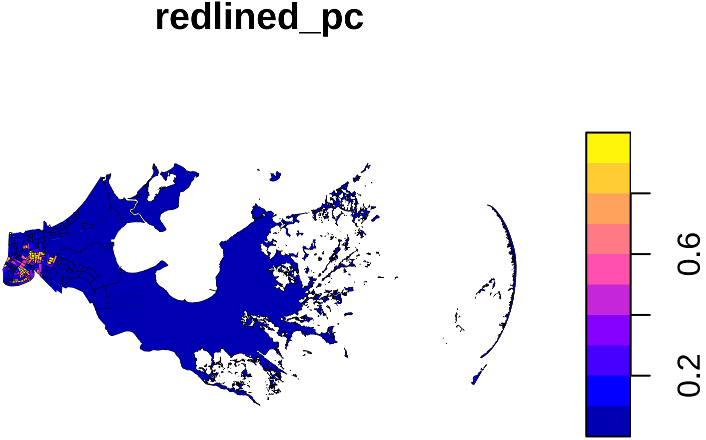
```
redlined_pc <-stats::aggregate(list(
  redlined_pc=as.numeric(tract2red$area_ix/tract2red$area)),
  by=list(GISJOIN=tract2red$GISJOIN),FUN=sum,na.rm=TRUE)
tracts2000 <- merge(tracts2000,redlined_pc,by="GISJOIN",all.x=TRUE)
```

Replace missing values with 0s:

```
tracts2000$redlined_pc[is.na(tracts2000$redlined_pc)] <- 0
```

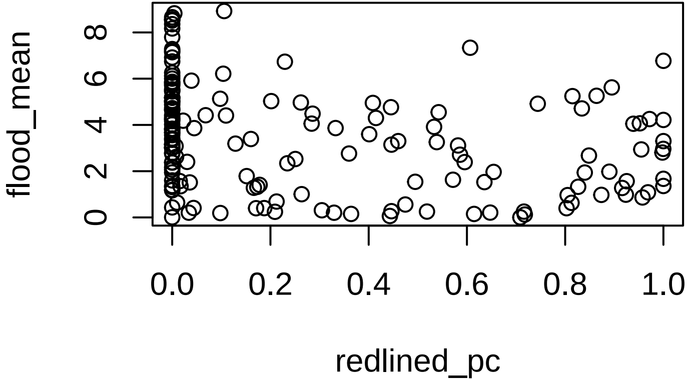
Inspect the results.

```
plot(tracts2000["redlined_pc"])
```



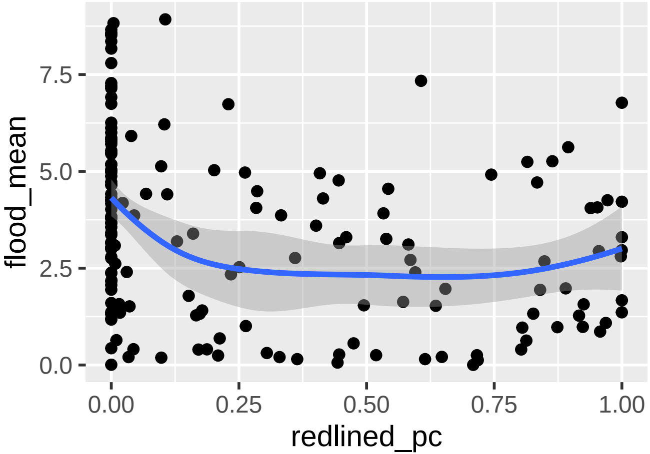
Make a *scatterplot* of flooding depth at different levels of redlining:

```
plot(x=tracts2000$redlined_pc,y=tracts2000$flood_mean)
```



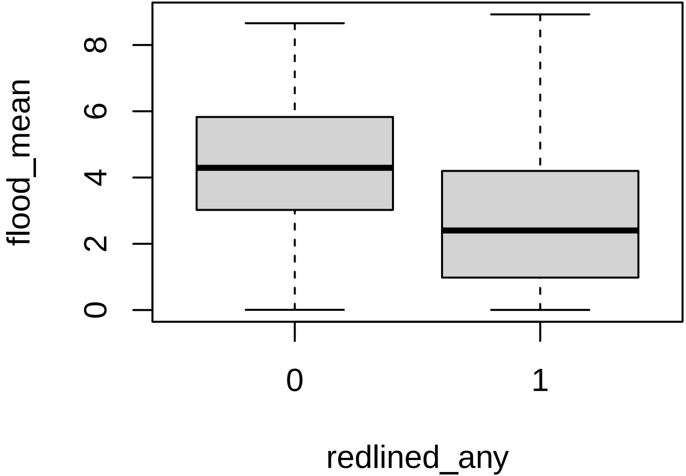
Make a *scatterplot* using ggplot2, with fitted LOESS curve:

```
ggplot2::ggplot(tracts2000, ggplot2::aes(x=redlined_pc,y=flood_mean))+
  ggplot2::geom_point()+
  ggplot2::geom_smooth()
```



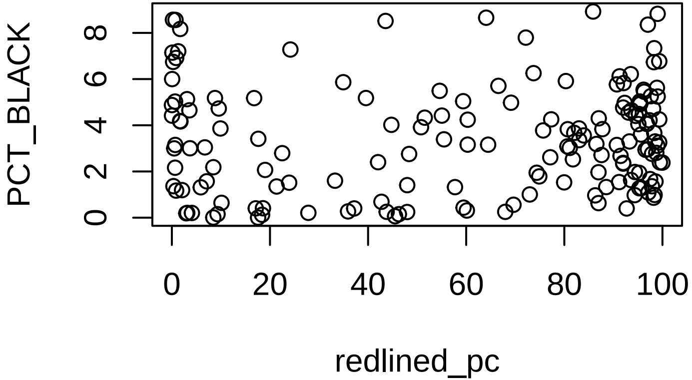
Make a *boxplot* of flooding depth with vs. without redlining:

```
tracts2000$redlined_any <- 1*(tracts2000$redlined_pc>0)
graphics::boxplot(flood_mean~redlined_any,data=tracts2000)
```



Make a *scatterplot* of race and redlining:

```
tracts2000$PCT_BLACK <- tracts2000$RACE_BLACK/tracts2000$POP_TOTAL*100
plot(x=tracts2000$PCT_BLACK,y=tracts2000$flood_mean)
```



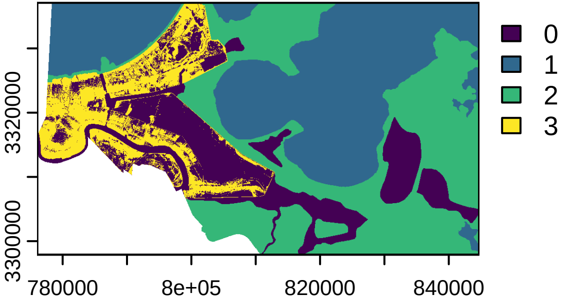
Raster reclassification

Load the *NFHL data* into R, and check CRS compatibility:

```
NFHL_NOLA <- terra::rast("Data/FEMA/NFHL_NOLA.tif")
sf::st_crs(NFHL_NOLA)==sf::st_crs(tracts2000)
```

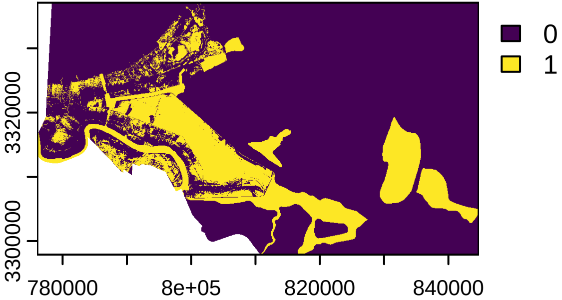
Plot the raster:

```
terra::plot(NFHL_NOLA)
```



Reclassify the NFHL raster to include just zone “AE” (0, high risk):

```
nfhl_highrisk <- 1*(NFHL_NOLA==0)
terra::plot(nfhl_highrisk)
```



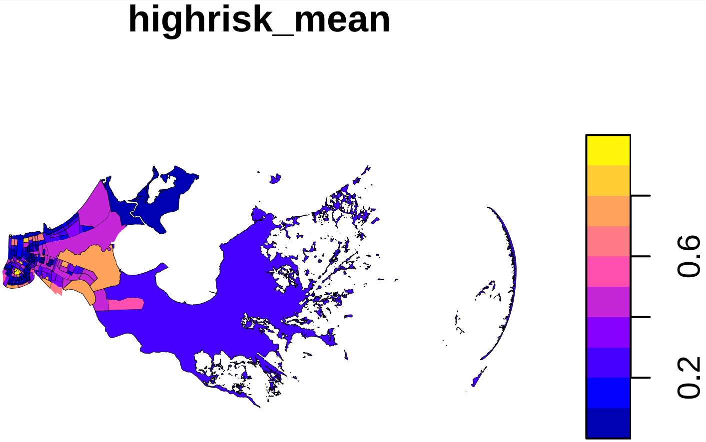
Calculate *zonal statistics*: proportion of each tract at high flood risk

```
tracts2000$highrisk_mean <- terra::zonal(x=nfhl_highrisk,
  z=terra::vect(tracts2000), fun=mean,na.rm=TRUE)[,1]
```

This may take a minute or two to run. . .

Inspect the results.

```
plot(tracts2000["highrisk_mean"])
```



Load the *311 data* into R, and convert to spatial points:

```
calls311 <- read.csv("Data/Call311/calls311_water_2012_2018.csv")
calls311 <- sf::st_as_sf(calls311,coords=c("longitude","latitude"),
  crs=4326)
```

Reproject the 311 data to same CRS as `tracts2000`:

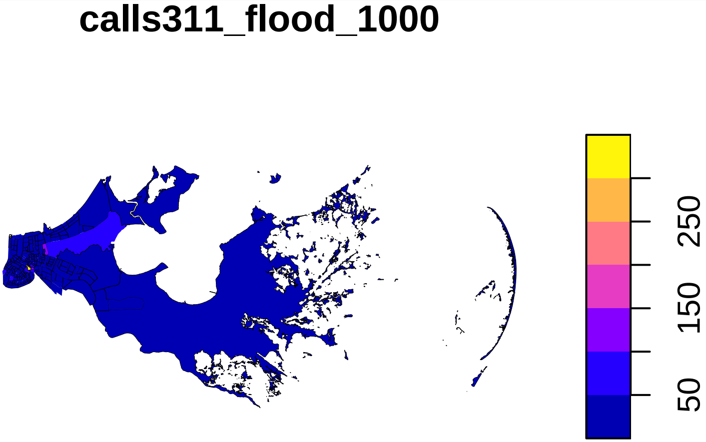
```
calls311 <- sf::st_transform(calls311,sf::st_crs(tracts2000))
```

Point-in-polygon analysis: 311 calls per 1000 residents, by type

```
tracts2000$calls311_flood_1000 <- lengths(sf::st_intersects(
  tracts2000,calls311[calls311$issue_type=="Street Flooding/Drainage",]
  )/tracts2000$POP_TOTAL*1000
tracts2000$calls311_basin_1000 <- lengths(sf::st_intersects(
  tracts2000,calls311[calls311$issue_type=="Catch Basin Maintenance",])
  )/tracts2000$POP_TOTAL*1000
tracts2000$calls311_mosquito_1000 <- lengths(sf::st_intersects(
  tracts2000,calls311[calls311$issue_type=="Mosquito Control",])
  )/tracts2000$POP_TOTAL*1000
```

Inspect the results.

```
plot(tracts2000["calls311_flood_1000"])
```



Problem Set 8

Your assignment (if using R): create two scatterplots

1. Flood risk and Katrina flood depth
   - `highrisk_mean` on x-axis
   - `flood_mean` on y-axis
   - no legend
   - axes and plot title properly labeled as on next slide
   - name the file `nfhl_katrina_R.png`
2. Flood risk and per capita 311 calls about street flooding
   - `highrisk_mean` on x-axis
   - `calls311_flood_1000` on y-axis
   - no legend
   - axes and plot title properly labeled as on next slide
   - name the file `nfhl_311flood_R.png`

- use either R base plots or `ggplot2`
- upload both plots to Canvas

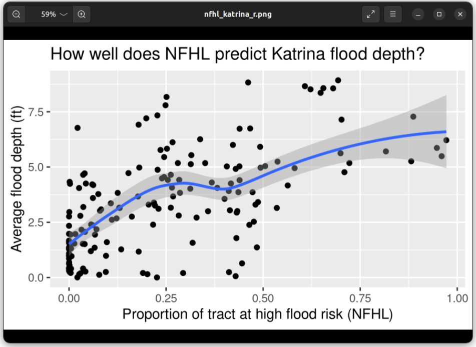


Figure 16: Can you make this?

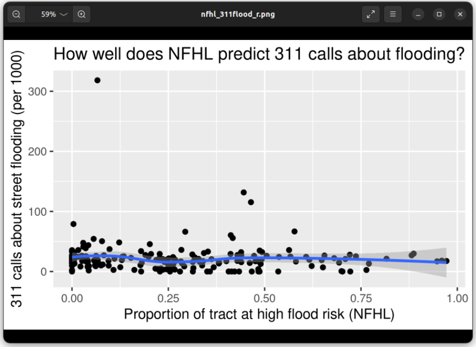


Figure 17: And this?